Universidad Autónoma de Madrid
Escuela politécnica superior

Grado en Ingeniería Informática

Trabajo Fin de Grado

CALIBRACIÓN AUTOMÁTICA DE MODELOS NEURONALES PARA SU USO EN CIRCUITOS HÍBRIDOS

Autor: Manuel Reyes Sánchez
Tutor: Pablo Varona Martínez

Junio 2016

CALIBRACIÓN AUTOMÁTICA DE MODELOS NEURONALES PARA SU USO EN CIRCUITOS HÍBRIDOS

Autor: Manuel Reyes Sánchez
Tutor: Pablo Varona Martínez

Grupo de Neurocomputación Biológica (GNB)
Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio 2016

Resumen

Este proyecto tiene como objetivo la calibración automática en tiempo real de modelos neuronales para su uso en circuitos híbridos, compuestos por neuronas vivas procedentes de sistemas biológicos y neuronas software. Estas conexiones se realizan por medio de sinapsis artificiales bidireccionales entre las neuronas vivas y las neuronas artificiales.

La calibración automática supone un paso importante en este tipo de conexiones dada la corta vida útil durante la cual pueden utilizarse las preparaciones de circuitos híbridos, apenas unas horas. Además los valores a fijar varían entre cada red biológica, por lo cual deben ser establecidos en cada experimento, siendo una tarea difícil que es posible y deseable evitar. También es interesante para estudiar los valores de conexión entre diferentes experimentos, o comparar como afectan diferentes factores en la conexión.

Los circuitos híbridos son de gran utilidad en el área de la investigación en neurociencia, ya que permiten caracterizar las dinámicas neuronales, así como las de las redes que conforman. También permiten mejorar el control de la actividad neuronal patológica, algo limitado por las técnicas neurofisiológicas tradicionales, lo cual puede ser incluso utilizado para combatir enfermedades neurológicas.

En los circuitos híbridos realizados se han utilizado neuronas pertenecientes a circuitos generadores de patrones del cangrejo de mar común, utilizando una conexión eléctrica para sincronizar el circuito vivo y el modelo.

Los modelos neuronales software utilizados poseen un comportamiento en ráfagas de disparo para la variable que representa el potencial de membrana. A la hora de realizar las conexiones ha sido necesario contar con las estrictas restricciones temporales que imponen estas conexiones por medio de la utilización de técnicas de ciclo cerrado, así como el escalado necesario entre modelos y neuronas vivas.

Para cumplir las condiciones de temporalidad se han explorado diferentes alternativas disponibles en una instalación típica del kernel Linux. También se ha utilizado la extensión RTAI, que proporciona tiempo real al kernel Linux. El envío y recepción de datos ha sido gestionado por medio de las librerías COMEDI, que permiten gestionar de manera transparente el uso de tarjetas de adquisición de datos.

El proyecto presenta algoritmos que consiguen escalar los modelos y señales así como determinar el valor de conductancia adecuado, obteniéndose una sincronización natural entre diferentes modelos y neuronas, incluyendo la validación realizada en un circuito híbrido.

Palabras Clave

Circuitos híbridos, modelos neuronales, neurociencia, sinapsis eléctrica artificial, ciclo cerrado, RTAI, COMEDI, software de tiempo real, pinzamiento dinámico, electrofisiología, CPG.

Abstract

The goal of this project is the automatic calibration in real-time of neural models to use them in hybrid circuits consisting of living neurons from biological systems and software neurons. These connections are made via two-way artificial synapses between living neurons and artificial neurons.

Automatic calibration is an important step in this type of connections because of the short period of time during which hybrid circuits preparations can be used, just a few hours. In addition, there is a large variability between each biological network which should be assessed in each experiment. This is a difficult task that can be avoided. It is also interesting to study the connection parameters between different experiments, or compare how different factors affect the connection.

These hybrid circuits are useful in the area of neuroscience research, allowing to characterize single neuron and neural network dynamics. They also allow better control of irregular activities, something limited by traditional neurophysiological techniques. This control can be useful in the context of some neurological diseases.

Neurons of a central pattern generator circuit from a common crab have been used to built hybrid circuits with an electrical connection to synchronize the living neurons and the models.

The used neural models have a spiking-bursting behavior for the variable that represents the membrane potential. The strict timing restrictions imposed by these connections have been taking into account using close loop techniques, as well as the necessary scaling between models and living neurons.

To meet the timing constraints different alternatives available in a typical installation of the Linux kernel have been explored. Also the RTAI extension have been used, which provides real-time to the Linux kernel. Sending and receiving data has been managed by the COMEDI libraries, which allow a transparent communication with the data acquisition cards.

The project presents algorithms that scale models and signals and determine the appropriate value of the coupling conductance. Synchronization between different models and neurons, including a validation performed with an hybrid circuit is achieved.

Key words

Hybrid circuits, neuronal models, neuroscience, artificial electrical synapse, closed loop, RTAI, COMEDI, real time software, dynamic clamp, electrophysiology, CPG.

Agradecimientos

Gracias a Mario, Jorge y al resto de compañeros de la carrera por la ayuda, por las noches de programación y por los buenos ratos en el comedor.

Y sobre todo muchas gracias a mis padres, gracias a su esfuerzo he llegado hasta aquí.

Índice general

Índice de Figuras

Índice de Tablas

Glosario

- **AEC**: Compensación Activa de Electrón (Active Electrode Compensation)
- **Burst**: Ráfaga de disparos neuronales
- **Cl^-**: Ión de cloro
- **COMEDI** :Control and Measurement Device Interface
- **CPG**: Generador Central de Patrones (Central Pattern Generator)
- **DAQ**: Adquisición de Datos (Data Acquisition)
- **DBS**: Estimulación cerebral profunda (Deep brain stimulation)
- **Despolarización**: Aumento del voltaje de membrana
- **ECM**: Error cuadrático medio
- **EDO**: Ecuación diferencial ordinaria
- **EPS**: Escuela Politécnica Superior
- **GNB**: Grupo de Neurocomputación Biológica
- **Hiperpolarización**: Disminución del voltaje de membrana
- **K^+**: Ión de potasio
- **Na^+**: Ión de sodio
- **RTAI**: Real Time Aplication Interface
- **Spike**: Disparo neuronal
- **TFG**: Trabajo Fin de Grado
- **UAM**: Universidad Autónoma de Madrid

1 Introducción

El siguiente capítulo detalla la motivación de proyecto, así como los objetivos perseguidos. También se indica la estructura de la presente memoria.

1.1. Motivación

Este Trabajo Fin de Grado se enfoca en el diseño e implementación de circuitos híbridos entre neuronas vivas y modelos artificiales con calibración automática de estas interacciones. Realizar estas conexiones es interesante pues ayuda a entender mejor el funcionamiento de las redes neuronales biológicas.

Desde comienzos del siglo XX se comenzó a investigar sobre el funcionamiento del sistema nervioso y en concreto sobre la fisiología neuronal (Ramón y Cajal, 1909). Esas primeras investigaciones permitieron definir a la neurona como la unidad básica del sistema, observando también el flujo de información que se produce entre ellas, así como otros factores como puede ser sus diferentes morfologías.

Actualmente conocemos mucha información sobres las neuronas, como que son unidades que generan actividad eléctrica de manera robusta y fiable, que cuentan con una gran plasticidad, adaptabilidad y tolerancia a fallos o que presentan un procesamiento de la información no lineal dependiente de la historia previa de estímulos (Rabinovich et al., 2006).

Típicamente las dinámicas neuronales solo han podido ser observadas parcialmente por medio del potencial de membrana, la concentración de calcio o los consumos de oxígeno. Los métodos de pinzamiento clásico (Chamorro et al., 2012) solo tienen en cuenta el potencial de membrana y por otro lado los sistemas neuronales son altamente no lineales, adaptativos y generalmente trabajan en régimen transitorio. A todo esto hay que añadir un problema de observación parcial. Por lo tanto los métodos por los cuales obtenemos información y estimulamos estos sistemas se encuentran muy limitados. En este complejo contexto, la interacción de bucle cerrado es una técnica que nos proporciona diversas posibilidades para caracterizar la dinámica con mediciones parciales, así como para realizar una estimulación dependiente de la actividad (Chamorro et al., 2012) con el objetivo de superar los obstáculos citados.

Esto requiere la utilización de modelos con un equilibrio entre realismo y complejidad que reproduzcan el potencial de membrana de una neurona real, así como de otros parámetros como

puede ser la concentración de calcio (Hindmarsh and Rose, 1984). Así mismo es necesario que los modelos acepten como entrada la señal obtenida, para conseguir una sincronización al sistema neuronal vivo.

Este trabajo trata de implementar una calibración automática de los factores de escala y conversión que afectan a la conexión. La importancia de la calibración viene dada por la necesidad de obtener una conexión correcta, consiguiendo incluir nuestro modelo neuronal en la red, pudiendo de esta manera observar la dinámica que sigue la red o comprobar cómo reacciona a estímulos mediante esta interacción.

Para realizar esta calibración deberán ser escaladas las señales enviadas y recibidas, pues cada modelo neuronal tiene rangos de funcionamiento diferentes a los cuales deben ser adaptados los datos que sean introducidos en él. Además de la escala de las señales será necesario calcular un valor adecuado de la conductancia de la conexión para conseguir una sincronización adecuada y controlar el tiempo de respuesta, pues las interacciones deben ser producidas con una frecuencia de tiempo estricta.

En este caso concreto, para la validación de las implementaciones se utilizarán circuitos generadores de patrones (Hooper, 2001) a los cuales serán conectados bidireccionalmente las neuronas artificiales por medio de sinapsis eléctricas. Estos circuitos están encargados de la generación de ritmos motores en los organismos y su estudio es interesante para caracterizar la negociación de estos ritmos, y en concreto del balance entre robustez y flexibilidad dinámica que poseen (Selverston et al., 2000). El estudio de estos circuitos en invertebrados ha permitido utilizar estos modelos de generación de ritmos en sistemas de osciladores neuronales acoplados bioinspirados en robots modulares para controlar su locomoción (Herrero-Carrón et al., 2011).

1.2. Objetivos

El objetivo de este proyecto es la autocalibración de modelos neuronales para conseguir una sincronización activa en circuitos híbridos. Por lo tanto se busca automatizar la elección de diversos parámetros de la conexión como son las escalas de las señales, los valores de conductancia de las conexiones o los ajustes necesarios para garantizar una correcta temporalidad.

Para ello se realizará un primer análisis del sistema nervioso, de las redes neuronales y de los modelos neuronales con comportamiento en ráfagas. Posteriormente ese conocimiento será utilizado para realizar circuitos híbridos por medio de técnicas de estimulación en ciclo cerrado.

El objetivo final es validar los algoritmos de autocalibración mediante la realización de circuitos híbridos, utilizando para ello circuitos generadores de patrones (CGP) biológicos.

Para cumplir con los objetivos de estricta temporalidad y conexión el proyecto debe utilizar herramientas que permitan cumplir los requisitos temporales y herramientas que permitan la comunicación con tarjetas de adquisición de datos. Estas herramientas son:

- Utilización de una Tarjeta de Adquisición de Datos (DAQ) para recibir y enviar datos, manejada por medio de las librerías COMEDI (`http://stm.lbl.gov/comedi/`).
- Estricta temporalidad para lo que se probarán diferentes alternativas, incluyendo el uso de un sistema operativo de tiempo real como es Linux con RTAI (`https://www.rtai.org`).

1.3. Estructura de la memoria

La presente memoria cuenta con los siguientes capítulos:

- **Estado del arte:** En este capítulo se exponen los conceptos teóricos necesarios para la realización del proyecto. Comenzando por nuestros conocimientos actuales del sistema nervioso se describe su principal unidad funcional, la neurona, y los procesos particulares de esta célula que son de interés para el proyecto, como la sinapsis, detallando el proceso y los factores que intervienen. Así mismo se listan diferentes modelos para reproducir comportamientos neuronales y se define en que consiste la estimulación neuronal en ciclo cerrado o los circuitos híbridos.
- **Hardware y software utilizado:** Se detallan las características de los equipos, como ordenadores y DAQ, utilizados en las diferentes fases del proyecto, así como el software externo utilizado (RTAI o Comedi).
- **Diseño y desarrollo:** Indica las características, descripción y motivo de elección de los modelos que se ha decidido utilizar. Se definen los parámetros a calibrar y los métodos que se seguirán para conseguir este objetivo. Incluye el resultado del análisis de problema realizado listando las tecnologías necesarias para el desarrollo del proyecto.
- **Experimentos realizados y resultados:** En este capítulo se indica el resultado de las bibliotecas desarrolladas, las primeras pruebas con modelos artificiales y se describen las validaciones del software realizado llevadas a cabo en primer término sobre una neurona electrónica y finalmente con un circuito híbrido.
- **Conclusiones y trabajo futuro:** En este punto se exponen las conclusiones del proyecto, el cumplimiento de los objetivos marcados y el análisis final del trabajo realizado. También se detalla el posible desarrollo que puede tener el trabajo en un futuro.
- **Anexos:** Detalles concretos sobre las soluciones de tiempo real, los modelos neuronales utilizados y las librerías desarrolladas.

2

Estado del arte

En este capítulo se detallan las bases de conocimiento necesario para el entendimiento del proyecto, como son el detalle del sistema nervioso, la neurona, los modelos neuronales, los circuitos híbridos y la estimulación neuronal en ciclo cerrado.

2.1. El sistema nervioso

Actualmente sabemos que el sistema nervioso esta fundamentalmente formado por dos tipos de células:

- **Neuronas:** Son las unidades básicas estructurales y funcionales del sistema nervioso. Son la unidad de estudio del presente proyecto. Son tratadas en detalle en este mismo capítulo (sección 2.2).
- **Células gliales:** Su función es la de soportar las neuronas, aunque también intervienen en el proceso cerebral de información. Estas células controlan el microambiente, esto es, su composición iónica, los niveles de neurotransmisores, el suministro de proteínas y otros factores de crecimiento (Gómez Nicola and Nieto Sampedro, 2008).

El presente conocimiento parte de los estudios de Ramón y Cajal, que definió la neurona como unidad funcional y básica del sistema nervioso (Ramón y Cajal, 1909), por medio de la técnica de tintado desarrollada por Camillo Golgi, que permitió un análisis celular incluso en tejidos de alta densidad como es el cerebral y por el cual ambos ganaron el premio nobel en el año 1906. Esta teoría definió el sistema nervioso como un conjunto de unidades independientes y discretas, las neuronas, que mediante su comunicación formaban una red (López-Muñoz et al., 2006).

Anteriormente a estos estudios se pensaba que el sistema nervioso estaba formado por un retículo continuo, no considerando las neuronas como células discretas e independientes que establecen conexiones, esto se conoce como teoría reticular, y dió paso a la doctrina de la neurona gracias a los experimentos de Ramón y Cajal (Sabbatini, 2003).

2.2. Unidad básica: la neurona

Las neuronas son unas células con un alto grado de especialización, que se encargan de la recepción de estímulos y en la conducción de impulsos nerviosos a través del sistema nervioso, interconectando sus distintos componentes. Se pueden realizar diferentes clasificaciones de los componentes, según nos centremos en topología, funciones u otras características. Generalmente podemos distinguir tres sistemas: sistema sensorial, sistema nervioso central y sistema motor. El desarrollo o proporción de cada uno de ellos puede variar entre especies, así mismo el número de neuronas que componen el cerebro también varía entre especies (Williams and Herrup, 1988) así como la proporción entre neuronas y células gliales.

2.2.1. Morfología de una neurona

Si bien las neuronas presentan una gran variedad de topologías todas tienes unas partes típicas, que pueden observarse en la figura 2.1.

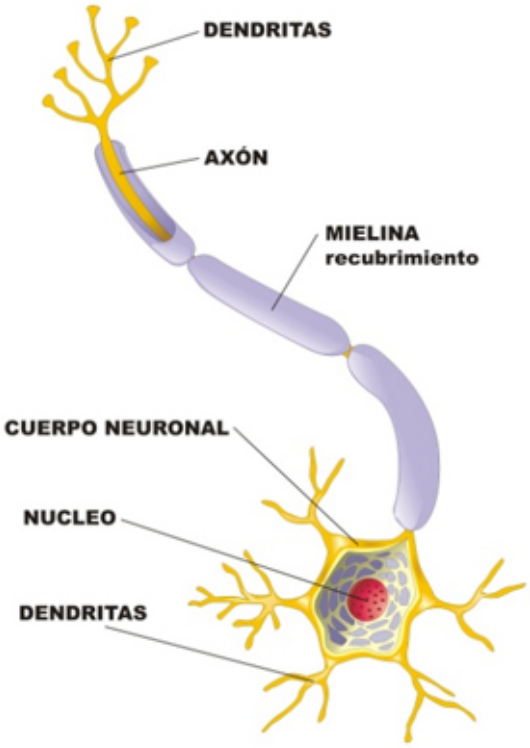


Figura 2.1: Representación de una neurona señalando sus partes más importantes.

- **Núcleo o soma:** contiene los orgánulos típicos y principales de toda célula, formando el cuerpo celular.
- **Dentritas:** son los receptores de señales nerviosas procedentes de otras neuronas o receptores sensoriales. Se establecen ramificadamente en forma de aboles.
- **Axón:** actúa como conducto de las señales producidas en el cuerpo celular hasta los botones terminales. Estos botones terminales se encuentran en las ramificaciones finales del axón y se encargan de liberar neurotransmisores, que se encargan de excitar o inhibir a las neuronas receptoras (Chen et al., 1997).

2.2.2. Membrana neuronal

Las neuronas se encuentran separas del medio donde se encuentran. Esto se consigue por medio de una membrana citoplasmática, que tiene de 6 a 8 nanomilímetros de grosor. La membrana presenta una permeabilidad selectiva, lo que permite la entrada y salida de determinadas moléculas. Estos canales pueden ser de dos tipos:

- **Canales iónicos pasivos:** siempre se encuentran abiertos.
- **Canales iónicos activos:** se abren o se cierran por medio de señales eléctricas, mecánicas o químicas.

La membrana neuronal posee una diferencia de potencial, llamada potencial de membrana, debido a que la carga eléctrica es opuesta en cada uno de los lados. La evolución en el tiempo de dicho potencial es usado para caracterizar el comportamiento de dicha neurona.

Cuando una neurona se encuentra en reposo y no recibe ningún estímulo el interior celular presenta una carga negativa respecto al exterior. Este valor puede variar, pero puede generalizarse como -65 mV, siendo el potencial de referencia exterior de 0 mV. Estando la neurona en reposo los canales activos suelen estar cerrados mayoritariamente (Hille, 1977; Kandel et al., 2000).

2.2.3. Sinapsis neuronal

La comunicación se produce por lo tanto por las conexiones entre axones y dentritas, concretamente por la generación de potenciales de acción o impulsos nerviosos gracias a la liberación de neurotransmisores. Esta conexión recibe el nombre de sinapsis.

La sinapsis queda definida por lo tanto como la unión funcional, intercelular y especializada entre neuronas (Hormuzdi et al., 2004). Podemos distinguir dos tipos principales de sinapsis:

- **Sinapsis eléctrica:** se produce entre neuronas que están próximas entre si, ya que esta cercanía permite que la corriente llegue directamente de una membrana a otra. Esto significa que no hay secreción de neurotransmisores, solo paso de iones entre las células por medio de uniones gap (un tipo de conexiones entre células).

  Esta comunicación se puede realizar simétricamente y simultáneamente en ambos sentidos con igual intensidad, produciendo una transmisión bidireccional. Sin embargo también puede producirse un flujo de corriente en un solo sentido, como ocurre en la sinapsis rectificadora. Comparada con una sinapsis química es muy rápida y no hay retardos en la comunicación (Kandel et al., 2000).

  Las uniones gap realizan una resistencia al flujo de corriente y esto es lo que determina si la corriente circula en ambos o solo en uno de los sentidos. La corriente que circula en cualquier caso es proporcional a la diferencia de los potenciales de las neuronas.

  Es el tipo de conexiones que se realizarán en el presente proyecto, concretamente serán transmisiones de tipo bidireccional, mediante el uso de modelos de la forma: $g_{21}(x_1(t) - x_2(t))$ en un sentido y $g_{12}(x_2(t) - x_1(t))$ en el sentido contrario, siendo x un potencial de membrana y g un valor de conductancia.

- **Sinapsis química:** este tipo de sinapsis es producida por los neurotransmisores que se encuentran en los botones terminales de los axones, en unas membranas llamadas vesículas sinápticas y que son liberados cuando se produce un potencial de acción (o momentos antes incluso).

  A diferencia de la sinapsis eléctrica esta puede ser gradual. Además este tipo de sinapsis es asimétrica, en estructura y funcionamiento, y no existe conexión entre las neuronas implicadas.

  Los neurotransmisores llegan a la neurona postsináptica y se unen a los receptores con los que esta cuenta, lo que ocasiona la apertura de los canales iónicos, ocasionándose así un flujo de iones que cambia el potencial de la segunda neurona. Existen neurotransmisores excitadores que despolarizan la neurona y existen también neurotransmisores inhibidores que la hiperpolarizan. Este tipo de sinapsis tiene un retardo en la transmisión que se sitúa en el orden de 1 a 5 milisegundos, realizándose la medida desde que se produce un potencial de acción en la neurona de origen hasta que se producen en la neurona destino (Kandel et al., 2000; López-Muñoz et al., 2006).

2.2.4. Potencial de membrana

Cuando se produce un movimiento de cargas eléctricas por los canales iónicos de la membrana celular se genera un potencial de acción conocido como potencial de membrana. Se dice que este potencial sufre una despolarización cuando aumenta y una hiperpolarización cuando disminuye.

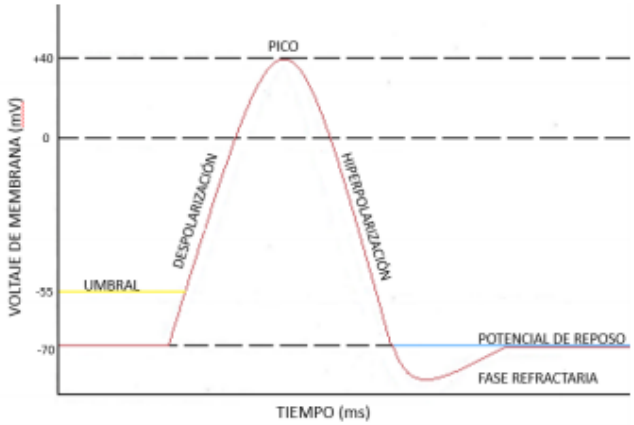


Figura 2.2: Ejemplo de proceso de activación y vuelta al estado de reposo de una neurona.

En el caso de que se supere el umbral de despolarización de la neurona esta producirá un potencial de acción (spike). Es normal que se produzcan periodos que alternan spikes y estados de reposo. Esta alternancia de estados es conocida como actividad spiking.

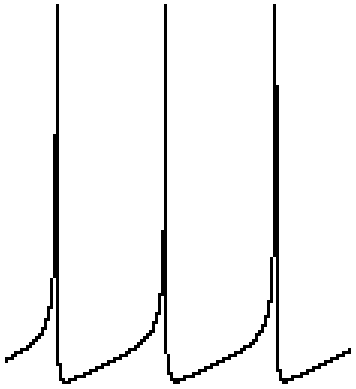


Figura 2.3: Actividad neuronal que alterna spikes y estados de reposo conocida como spiking.

También se producen agrupaciones de potenciales de acción en muy cortos intervalos de tiempo. Cuando esto ocurre se le denomina con el nombre de ráfagas (burst). Al igual que la actividad spiking, es común observar este comportamiento en la actividad neuronal, y es conocido como actividad bursting (Kandel et al., 2000).

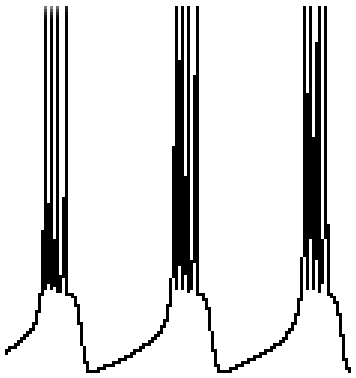


Figura 2.4: Actividad neuronal que alterna agrupaciones de spikes y estados de reposo conocida como bursting.

2.3. Modelos neuronales

2.3.1. Modelo electrónico

Habitualmente dentro de las células neuronales encontramos mayores concentraciones de iones de potasio K^+ y fuera iones de sodio Na^+ y cloro Cl^-. La mencionada membrana neuronal (véase apartado 2.2.2) regula el flujo de estos iones, generando con ello corrientes eléctricas. Cada ión recorre sus canales iónicos específicos, teniendo que vencer cada uno su potencial concreto.

Para representar esto Cole y Curtis (Cole and Curtis, 1939) propusieron, basándose en sus observaciones experimentales del axón gigante del calamar, un circuito para representar el comportamiento de estas membranas.

Este circuito (ver figura 2.5) pretende representar las propiedades de la membrana neuronal. Las resistencias equivalen a los canales iónicos, las fuentes de tensión actúan como los gradientes de concentración de los diferentes iónes y el condensador representa la capacidad de almacenamiento de carga eléctrica que posee la membrana.

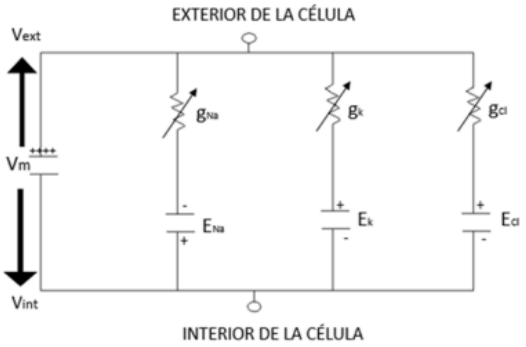


Figura 2.5: Circuito análogo al comportamiento pasivo de una membrana neuronal con canales de K^+, Na^+ y Cl^-.

2.3.2. Modelos matemáticos

La equivalencia que se puede realizar entre los procesos de la membrana neuronal y un circuito electrónico (apartado 2.3.1) permite realizar modelos matemáticos que describan la dinámica del voltaje de la membrana neuronal en el tiempo. También es posible que estos modelos reflejen otros comportamientos, dependiendo de su realismo, como puede ser la concentración de un cierto ión.

Los primeros trabajos de modelos neuronales surgieron desde una perspectiva del procesamiento de la información que se produce en el sistema nervioso (Mcculloch and Pitts, 1943). Durante la misma década también comenzaron a desarrollarse reglas de aprendizaje basadas en procesos neuronales (Hebb, 1949).

Por otra parte en el campo de la física se comenzó a estudiar las membranas de gran variedad de células utilizando en particular el axón gigante del calamar. Esto llevó a observar que la conductancia de las membranas celulares aumentaba en gran cantidad al producirse un potencial de acción (Cole, 1968; Cole and Curtis, 1939).

La suma de todos estos trabajos llevó a los neurofisiólogos A. Hodgkin y A. Huxley a realizar más experimentos con el axón gigante de calamar, midiendo los potenciales extracelulares que se producían debido al paso de un potencial de acción, utilizando la diferencia de potencial entre una región despolarizada y otra en reposo. Consiguieron con ello un modelo matemático para representar el potencial de membrana de una neurona ante una corriente de iones inyectada (Hodgkin and Huxley, 1952).

Los llamados modelos de Hodgkin-Huxley estaban centrados en representar el comportamiento real, conteniendo una ecuación para el potencial y una ecuación por cada una de las diferentes conductancias. En las décadas posteriores surgieron una serie de modelos, que pueden ser considerados a grandes rasgos variaciones de los presentados por Hodgkin y Huxley (Connor et al., 1977).

Un ejemplo de estos modelos es el modelo FitzHugh Nagumo (Nagumo et al., 1962). Dos décadas mas tarde se publicó el modelo Morris-Lecar (Morris and Lecar, 1981), desarrollado de acuerdo a la fibra muscular gigante del percebe y probado por ejemplo en neuronas del ganglio estomatogástrico de la langosta (Skinner et al., 1993) o neuronas sensoriales de mamíferos (Prescott et al., 2008). También se han presentado esquemas para reducir el número de ecuaciones diferenciales de los modelos, como el modelo Kepler, con ejemplos de su funcionamiento sobre modelos ya existentes (Kepler et al., 1992).

Sin embargo el modelo que presento un gran cambio fue el presentado por Hindmarsh y Rose, que modificaron el modelo FitzHugh Nagumo, sustituyendo la función lineal por una cuadrática.

Este modelo incorpora tres ecuaciones diferenciales no lineales (Hindmarsh and Rose, 1984).

Durante el siglo XXI han surgido modelos que avanzan en la linea de ser muy versátiles, disminuyendo además la complejidad y el detalle del proceso real. Esto presenta ventajas como la necesidad de una menor capacidad computacional o poder centrarse en otros aspectos si el interés no son las reacciones químicas si no el procesamiento y la transmisión de la información.

El modelo desarrollado por Rulkov plantea el uso de un mapa iterado en dos dimensiones, con una señal dinámica rápida y otra de dinámica lenta (Rulkov, 2002), lo que sirve para realizar con mayor simplicidad actividad en ráfagas (véase figura 2.4).

Por su parte Izhikevich ideó un modelo con dos ecuaciones diferenciales, consiguiendo un modelo simple pero que puede reproducir el comportamiento de una neurona real (Izhikevich, 2003), reduciendo enormemente el coste computacional respecto al modelo Hodgkin-Huxley, lo que es de gran ayuda para simulaciones con un gran numero de neuronas.

En este proyecto se han seleccionado modelos con un comportamiento en ráfagas, para que puedan ser utilizados en circuitos híbridos con CPGs, por medio de una comunicación bidireccional, como son el modelo de Hindmarsh y Rose, el mapa de Rulkov o el modelo de Izhikevich.

2.4. Estimulación neuronal en ciclo cerrado

La estimulación en ciclo cerrado permite caracterizar la dinámica de la actividad registrada así como controlar la actividad normal. La utilización de estos métodos está justificada por ser los sistemas nerviosos no lineales y adaptativos y procesar la información en régimen transitorio (Chamorro et al., 2012).

Estos métodos presentan una gran variedad de posibilidades para estudiar diferentes propiedades neuronales. Por ejemplo el método de fijación de voltaje (voltage clamp) consiste en la fijación del voltaje para medir la corriente de iones a través de la membrana (Destexhe and Bal, 2009; Roth et al., 2014) pudiendo así definir los comportamientos de estas corrientes. Por medio de estos mismos sistemas es posible añadir modelos artificiales al sistema (véase 2.3.2), que pueden interactuar con las neuronas vivas.

En general estos conceptos han sido desarrollados, manteniéndose los principios, en nuevos métodos útiles para distintos contextos (Roth et al., 2014; Van Boxtel and Gruzelier, 2014). Por ejemplo pueden utilizarse para estudiar los ritmos o dinámicas de procesos neuronales y controlar estados patológicos o naturales que no son visibles en los protocolos anteriores de ciclo abierto. La precisión temporal por otro lado aporta un control y representación más eficaz de distintos parámetros como pueden ser la periodicidad, duración o intensidad del estímulo (Chamorro et al., 2012; Fernandez−Vargas et al., 2013; Potter et al., 2014).

2.5. Circuitos híbridos

Gracias a la técnica de Dynamic Clamp es posible acoplar los modelos neuronales a neuronas vivas por medio de conductancias sinápticas, estableciendo circuitos híbridos. Los primeros circuitos híbridos investigados (Yarom, 1991) consistieron en la conexión de una célula olivar y cuatro unidades oscilantes, con el objetivo de estudiar como las oscilaciones en estas neuronas biológicas provocan oscilaciones sincronizadas en el potencial de membrana de las diferentes neuronas.

Durante los años siguientes se produjeron más experimentos, observando por ejemplo la relación entre la reducción de la sinapsis y la biestabilidad del sistema (Manor and Nadim,

2001). También se han realizado estudios sobre fenómenos como la retroalimentación en circuitos del tálamo, a través del cual la información sensorial llega a la corteza cerebral (Le Masson et al., 2002). En este último estudio la simulación del circuito tálamo-cortical con un circuito híbrido permitió, mediante el control de ganancia de la inhibición de la retroalimentación, y la modulación de la excitabilidad de la membrana, conseguir la coordinación entre los circuitos talámicos para conseguir un correcta transmisión de la información desde los lugares de recepción de los estímulos sensoriales a la corteza cerebral.

Otro ejemplo de resultados obtenidos con estas técnicas son demostrar que propiedades como forma y amplitud del potencial de campo se encuentran relacionadas con las propiedades físicas de la neurona y su morfología (Cohen et al., 2006), lo que denota la importancia de la comprender los mecanismos que generan estos potenciales de campo para la implementación de neuronas electrónicas. Las neuronas utilizadas pertenecían a un molusco, observando que el potencial de campo es el resultado del flujo de la corriente longitudinal entre los compartimento de la neurona eléctricamente distantes.

También está comenzando a surgir software que permite superar barreras y limitaciones de los experimentos con circuitos híbridos y técnicas de pinzamiento dinámico, lo que permite llegar a un rango de usuarios mayor, como StdpC (Nowotny et al., 2006), ayudando a establecer estas técnicas como herramientas habituales en la electrofisiología. Este software presenta características de generación de onda, observación de la actividad y ha permitido avances como la creación de un nuevo método de compensación de electrodo activo (AEC).

2.6. Circuitos Generadores Centrales de Patrones

Los CPG son redes neuronales que generan ritmos motores. La mayoría de sus neuronas son motoneuronas que se encargan de comandar músculos para la realización de movimientos repetitivos, como puede ser la locomoción, la respiración o la masticación (Harris-Warrick, 2011). Estas redes generan comportamientos que son relativamente simples, pero aun así cuentan con gran flexibilidad para adaptar su comportamiento de acuerdo a las demandas del entorno, es decir sus neuronas negocian los ritmos para cada circunstancia.

Estos circuitos ya han sido estudiados acoplando dos neuronas vivas con una una conexión artificial (Elson et al., 1998) y o neuronas vidas del tipo PD de un CPG pilórico y neuronas artificiales (Szücs et al., 2000; Pinto et al., 2000), perteneciente a una langosta, mostrando que por separado estas neuronas muestran un comportamiento caótico pero una vez conectadas realizan ritmos regulares. El estudio también muestra la flexibilidad de estos ritmos. El citado proyecto es un ejemplo de como el patrón de actividad puede ser modificado por medio de sinapsis artificiales.

Siguiendo con el mismo circuito de la langosta, han sido desarrollados modelos concretos (Varona et al., 2001) con la intención de analizar los comportamientos caóticos que presentan por separado esas neuronas. Más recientemente (M. Hooper et al., 2015) se han realizado pruebas con diferentes neuronas pertenecientes a los CPG como son las neuronas AB, PD y LP. Mediante la estimulación a una neurona PD, con ráfagas análogas a las producidas por una neurona LP de la red pilórica, observando la dependencia existente entre el ritmo y la fase. Esto permitió observar cómo se producen mínimas variaciones a intervalos constantes.

El primer circuito híbrido con un CPG (Szücs et al., 2000), como el del presente proyecto, sirvió para conseguir reparar ritmos del CPG que habían sido modificados por un daño producido al circuito. El uso de estos circuitos ha demostrado ser eficaz, con aplicaciones en otros campos, como por ejemplo la implementación de osciladores neuronales acoplados a robots para controlar su locomoción (Herrero-Carrón et al., 2011).

2.7. Necesidad de la calibración automática

Controlar los parámetros de la conexión sea de manera automática o manual es en cualquier caso indispensable, debido a diversos motivos, siendo los más importantes las diferencias de escalas entre los diferentes modelos y las señales reales así como definir las conductancias necesarias para conseguir una sincronización activa. Los detalles sobre los parámetros a tener en cuenta, los métodos de medida e implementaciones pueden ser consultados más en detalle en el capítulo 4.2 y en el Anexo B.

Más allá del simple hecho de conseguir sincronizar correctamente nuestro modelo neuronal a la red biológica y el ahorro de tiempo que conlleva, la calibración tiene diversas utilidades como puede ser la detección automática de eventos o, conseguir una determinada periodicidad en las ráfagas (Stigen et al., 2011). En el citado artículo se muestra un algoritmo que consigue determinar la fase y amplitud necesarios para conseguir que una neurona dispare en un momento deseado.

Controlar las frecuencias de disparo y la sincronización también es útil para el tratamiento de diversas enfermedades neurológicas. Por ejemplo la sincronización neuronal juega un papel importante en dolencias como el Parkinson y la epilepsia (Benabid et al., 2009; Berenyi et al., 2012).

Un posible uso de las calibraciones consiste en su utilización en protocolos de simulación en ciclo cerrado sirviendo para sincronizar o desincronizar poblaciones neuronales dentro de estimulaciones cerebrales profundas (DBS). Este método consiste en la implantación de un aparato que envíe pulsos al cerebro, con casos de éxito que han conseguido mitigar las dolencias del Parkinson.

3

Hardware y software utilizado

Este capítulo tiene como objetivo detallar las características del material utilizado.

3.1. Hardware

3.1.1. Ordenadores

Para el desarrollo de este proyecto se han probado un conjunto de ordenadores para determinar si los modelos implementados pueden cumplir las restricciones temporales para ser utilizados en circuitos híbridos. Los requisitos generales (una ráfaga en un segundo) se cumplen incluso en ordenadores de gama baja, sin embargo para requisitos más concretos (necesidad de proporcionar y leer un punto cada 1/10.000 segundos) se ha comprobado la necesidad de contar con software que proporcione tiempo real estricto. Pueden comprobarse las soluciones utilizadas y los resultados obtenidos en el Anexo B.

Las características del ordenador utilizado en el laboratorio de GNB, con el que han sido llevadas a cabo las pruebas con neuronas electrónicas y vivas son las siguientes:

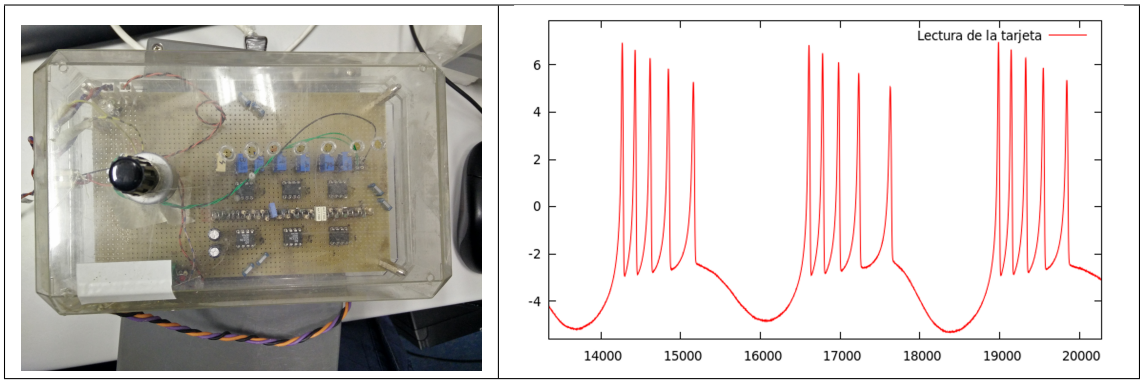
- **Sistema operativo:** Ubuntu Linux 10.04 LTS
- **Versión del nucleo Linux:** 2.6.35.7-rtai
- **Procesador:** Intel(R) Pentium(R) 4 CPU 3.20GHz, 32 bits, 4 núcleos

3.1.2. Tarjeta de Adquisición de Datos (DAQ)

- **Modelo de la tarjeta:** National Instruments PCI-6529
- **Rango de voltajes admitido:** 10, 5, 2, 1, 0.5, 0.2, 0.1 voltios

3.1.3. Neurona electrónica

El laboratiorio del GNB cuenta con una neurona electrónica (véase 2.3.1) que implementa el modelo de Hindmarsh-Rose (véase 4.1.1).



Cuadro 3.1: Fotografía de la neurona electrónica del GNB y señal que produce capturada por medio de la tarjeta DAQ.

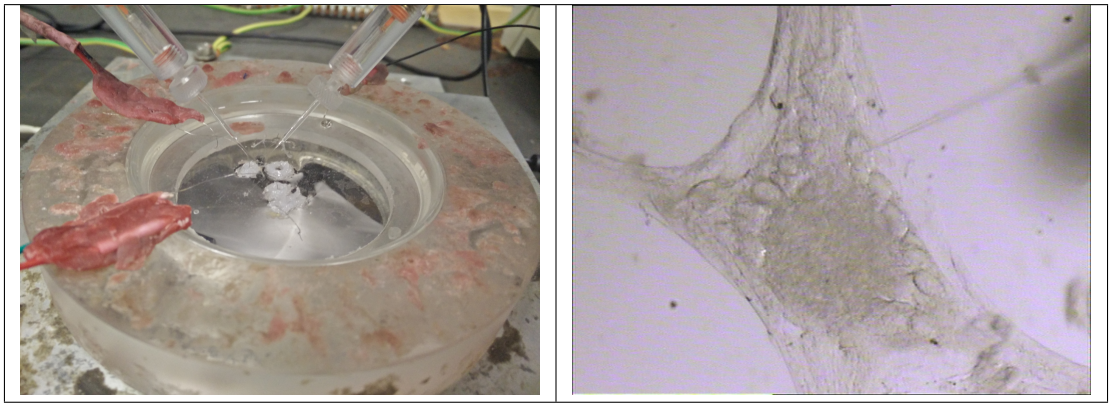
3.2. Software

Versiones de los programas instalados en el ordenador del laboratorio del GNB:

- **Driver Comedi utilizado:** ni_pcimio
- **Versión de Comedi:** 0.7.76
- **Versión de Comedilib:** 0.7.22
- **Versión de RTAI:** 3.7.1

3.3. Neuronas vivas

Las redes utilizadas pertenece a cangrejos de mar común. En concreto se han utilizado motoneuronas del CPG (véase 2.6) del ganglio estomatogástrico. En el cuadro 3.2 puede observase la preparación y una imagen obtenida en la que se ven las neuronas del CPG.



Cuadro 3.2: Preparación in vitro utilizada para la realización de un circuito híbrido e imagen de las neuronas utilizadas.

4

Diseño y desarrollo

En este capítulo se detallan los modelos neuronales que van a ser utilizados para la realización de los circuitos híbridos y se especifica cómo serán calibradas las conexiones. También incluye descripciones de las soluciones a utilizar para cumplir los requisitos de temporalidad y las librerías a desarrollar para encapsular el uso de Comedi o RTAI.

4.1. Modelos neuronales

De la lista de modelos expuestos (véase capítulo 2) se han elegido los modelos más versátiles, y más ligeros en carga computacional.

Aquí se exponen los modelos de manera general, para obtener información detallada sobre sus posibles comportamientos, detalles concretos de las implementaciones, valores utilizados o el método utilizado para resolver numéricamente ecuaciones diferenciales ordinarias consultar el Anexo B.

4.1.1. Modelo Hindmarsh-Rose

Este modelo está formado por tres ecuaciones que representan el potencial de membrana (x), una corriente rápida (y) y una corriente lenta (z) respectivamente (Hindmarsh and Rose, 1984). Su publicación marco una diferencia dado que tiene una estructura más sencilla que modelos anteriores (detallados en el punto 2.3), cambiando la función lineal por una cuadrática, y presentando un amplio rango de comportamientos (Szücs et al., 2000). También cabe destacar que este modelo permite realizar un comportamiento regular o caótico, esto se regulado por medio del parámetro I y siendo la frontera $I = 3{,}281$. Los parámetros a, b, c, d, f, μ, S, h son parámetros ajustables que modifican el comportamiento del modelo.

$$\frac{dx(t)}{dt} = ay(t) + bx(t)^2 - cx(t)^3 - dz(t) + I$$
$$\frac{dy(t)}{dt} = e - fx(t)^2 - y(t)$$
$$\frac{dz(t)}{dt} = \mu(-z(t) + S(x(t) + h))$$

El método de integración y los valores particulares iniciales utilizados se pueden observar en el Anexo B, así cómo otros detalles particulares de la implementación utilizada. Para resolver el sistema de ecuaciones diferenciales se utiliza el método de Euler.

4.1.2. Mapa de Rulkov

Este modelo es el más simple matemáticamente de todos los expuestos, cuentas con dos variables y permite representar tres regímenes de actividad (silencioso, disparo tónico y ráfaga tónica) (Rulkov, 2002) que pueden ser consultados en el Anexo B.

En el modo de ráfagas también es posible regular comportamientos cómo la cantidad de spikes o la frecuencia.

Las ecuaciones son las siguientes:

$$x_{n+1} = f(x_n, y_n + eI_n)$$
$$y_{n+1} = y_n - \mu(x_n + 1) + \mu\sigma$$

Contando con la siguiente función:

$$f(x,y) = \begin{cases} \frac{\alpha}{1-x} + y & si \quad x \leq 0 \\ \alpha + y & si \quad 0 < \alpha < \alpha + y \\ -1 & si \quad x \geq \alpha + y \end{cases}$$

En este caso la resolución de los sistemas de ecuaciones es directo.

4.1.3. Modelo Izhikevich

Este modelo (Izhikevich, 2003) permite imitar los comportamientos de diferentes neuronas de acuerdos a sus parámetros (consultar Anexo B para más información). Cuenta con dos ecuaciones que representan el potencial de membrana y la recuperación de la membrana respectivamente.

$$\frac{dv(t)}{dt} = 0{,}004v^2 + 5v + 140 - u + I$$
$$\frac{du(t)}{dt} = a(bv - u)$$

Este modelo cuenta además con una condición de reinicio dependiente del potencial. Típicamente se sitúa este valor en 30.

$$\text{Si } v >= 30 => \begin{cases} v = c \\ u = u + d \end{cases}$$

Para resolver este sistema de ecuaciones diferenciales ordinarias se ha utilizado el método de Euler.

4.2. Calibración automática

El principal objetivo de este TFG es calibrar automáticamente los parámetros que pueden afectar a la conexión entre un modelo neuronal y una neurona viva. Durante esta sección se habla de conexiones neuronales no importando si nos referimos a dos modelos conectados, un modelo y una neurona electrónica o un modelo y una neurona real.

La necesidad de calibrar la señal de los modelos surge debido a que el potencial de membrana de un modelo a es recibido por el modelo b. Lo mismo sucede simétricamente. Por lo tanto, se requiere que la señal enviada en ambos sentidos sea adecuada al modelo que la recibe, pues si no o bien no tendrá influencia (si es mucho más pequeña) o resultará demasiado determinante conduciendo al modelo afectado a una sincronización pasiva (si es mucho más alta). Además al trabajar con neuronas vivas el envío de un potencial elevado puede afectar a la neurona.

Por lo tanto, debe escalarse la señal del modelo a para ser enviada al modelo b y también debe ser escalada la señal del modelo b cuando sea enviada al modelo a.

Una vez realizada esta escala aún nos queda definir la conductancia. La conductancia multiplica a la señal de los modelos para definir la importancia de la señal recibida respecto a los propios parámetros del modelo.

4.2.1. Calibrar amplitud de las señales

Los modelos neuronales a implementar, como el modelo de Izhikevich, cuentan con unidades adimensionales, que pueden variar enormemente entre modelos.

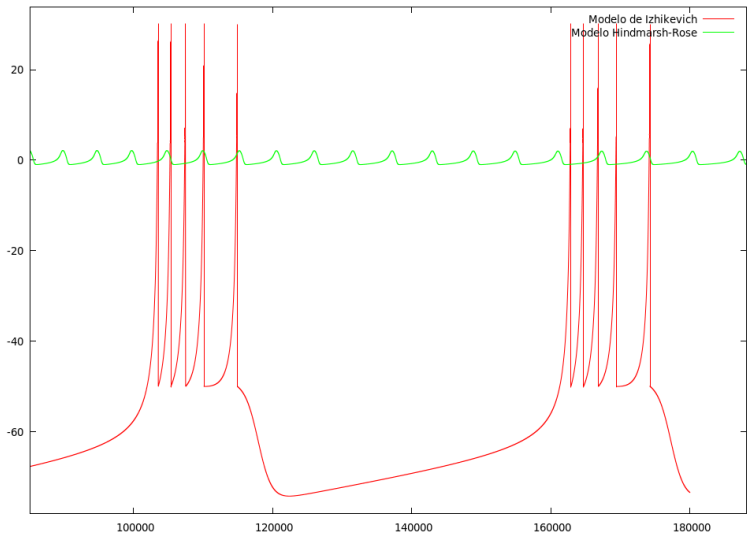


Figura 4.1: Comparación de las señales de los modelos de Izhikevich y Hindmarsh-Rose en la que se puede ver la diferencia de amplitud de las señales producidas.

Para realizar el escalado del ampliado de una señal A a una señal B se ha diseñado el siguiente algoritmo:

- Obtener el máximo y el mínimo de ambas señales
- Calcular rango de cada señal (máximo-mínimo)
- Escala A hacia B = Rango B / Rango A

4.2.2. Calibrar el offset vertical de las señales

Que los modelos cuenten con una amplitud similar (apartado 4.2.1) no garantiza que se encuentren en el mismo rango de valores, por lo tanto este comportamiento también debe ser ajustado.

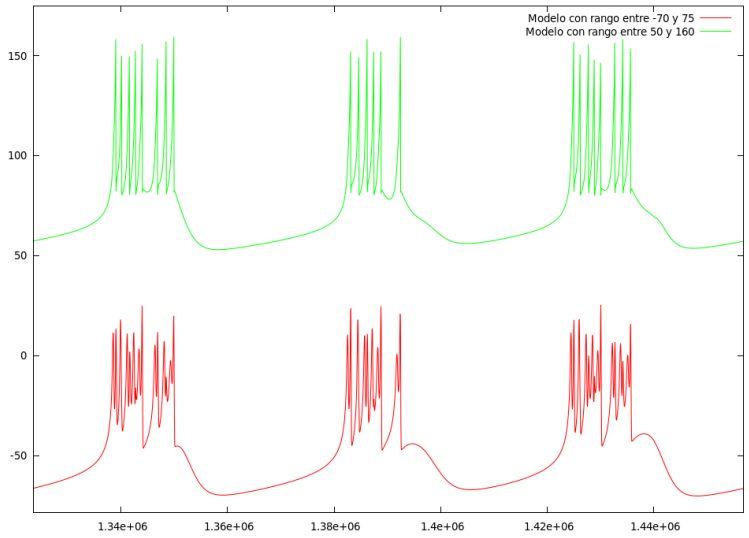


Figura 4.2: Imagen de dos señales con una amplitud similar pero que se encuentran en un rango de valores diferente.

El algoritmo que se seguirá para determinar el offset de un modelo A respecto a un modelo B es el siguiente:

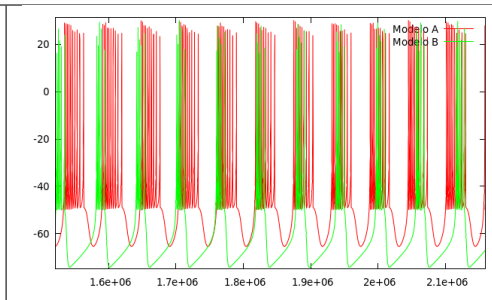
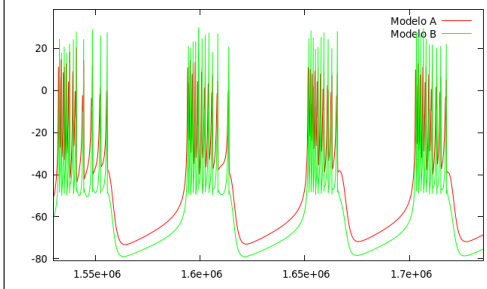
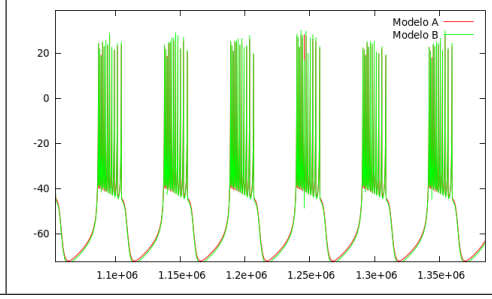
- Obtenemos las escalas de amplitud (véase 4.2.1)
- Offset modelo A hacia B = Mínimo B - (Mínimo A * Escala A hacia B)

4.2.3. Calibrar conductancia de la conexión

Contar con señales de escala similar no garantiza la sincronización de la conexión. Una vez los potenciales se encuentran en similar rango es necesario ajustar la fuerza de la conexión. Esto se hace por medio de un valor llamado conductancia y que se encarga de establecer como influye una señal en otra. Cuanto mayor es un valor de conductancia, más voltaje procedente de un modelo se inyecta en el modelo receptor. Esta conexión puede ser expresada de la forma $g_{21}(x_1(t) - x_2(t))$ en un sentido y $g_{12}(x_2(t) - x_1(t))$ en el sentido contrario, siendo x_1 el potencial de membrana de una neurona (viva o artificial), x_2 el potencial de la otra neurona y $g_(12)$ el valor de la conductancia en cada sentido.

Este valor debe ser ajustado para conseguir una sincronización adecuada. Con un valor bajo no habrá sincronización y con un valor alto la sincronización será pasiva, obligando una neurona a la otra a imitar sus potenciales sin existir una negociación del ritmo real.

Un valor infraestimado se observa en indicadores como la frecuencia de las ráfagas o el offset relativo entre las ráfagas. Un valor sobreestimado en la conductancia puede ser observado cuando ambos potenciales son continuamente muy parecidos, por ejemplo en sincronizaciones activas las fases refractarias (véase figura 2.2) no tienen porque coincidir.

Sin sincronización	
Sincronización activa	
Sincronización pasiva	

Cuadro 4.1: Cuadro comparativo entre dos neuronas no sincronizadas con diferentes frecuencias de disparo, dos neuronas con una sincronización buena y dos neuronas con sincronización pasiva.

Para calibrar el valor de la conductancia de la conexión se han llevado a cabo dos maneras diferentes de medir la sincronización:

Un primer método se ha basado en medir el offset entre ráfagas y la frecuencia de ambas ráfagas. Podemos aumentar progresivamente el valor de la conductancia hasta que la frecuencia de las ráfagas observadas sea la misma. Respecto al offset entre ambas ráfagas debemos establecer una tolerancia que puede ser ajustada y refinada mediante pruebas. La dificultad de estos métodos surge en cómo medir la frecuencia y offset de ondas que pueden estar cambiando su forma y rangos de actuación debido a la propia interconexión que se está produciendo entre el modelo y la neurona viva.

Adicionalmente el ruido y las formas más cambiantes de las señales biológicas son factores que pueden afectar a la hora de medir estos valores y que debe ser tenido en cuenta.

Como segundo método se ha establecido un criterio de sincronización dependiente de la diferencia de potenciales entre ambas señales. Para llevar a cabo este cálculo se establecerá la diferencia absoluta entre cada punto. El sumatorio de estos valores podrá ser utilizado para definir el grado de sincronización, detectando así si se debe aumentar el valor de conductancia.

Para ambos métodos deberá establecerse una ventana de observación, tras la cual se evalúen los resultados y se decida el grado de sincronización. Esto es debido a que tras establecer un nuevo valor de conductancia debemos observar como afecta a la conexión. Esta ventana deberá ser definida experimentalmente, buscando encontrar un equilibrio entre valores que permitan medir adecuadamente y establecer medias adecuadas y no tardar mucho en realizar el proceso, pudiendo situarse aproximadamente en valores entre 2 y 4 ráfagas.

4.3. Requisitos estrictos de temporalidad

La comunicación en tiempo real requiere escribir y leer datos en la tarjeta de aquisición de datos de acuerdo a una frecuencia, que en este proyecto estará fijada a 10kHz (aunque este valor puede ser variado), debiendo tener listo el valor a enviar cada 1/10.000 segundos.

En el proyecto las ráfagas de los modelos neuronales serán definidas automáticamente por medio de los datos leídos por el modelo, estableciendo las ráfagas del modelo a igual frecuencia. De acuerdo al número de puntos producidos por un modelo en una ráfaga deberán tomarse las medidas oportunas para controlar este valor, pudiendo una cantidad de puntos calculada automáticamente no ser enviados por la tarjeta con el objetivo de conseguir que una ráfaga dure el tiempo fijado y además se envíe un punto calculado en el momento determinado.

Durante la implementación y primeras pruebas se utilizarán métodos para conseguir la estricta temporalidad que son fáciles de utilizar en una instalación de linux. Estos métodos son además una manera más pedagógica de aprender puesto que requieren un mayor esfuerzo en el lado de la lógica desde el punto de vista de la programación.

Estos métodos no garantizan sin embargo una estricta temporalidad como la que aporta un sistema con tiempo real como el que se utilizará en las pruebas finales. Estos sistemas cuentan con el problema de necesitar instalaciones que no son típicas en los ordenadores y son mas complicados de instalar. Sin embargo cuentan con una mayor precisión y mayor facilidad para establecer la periocidad de las tareas.

Aprovechando el uso de diferentes soluciones se ha realizado un pequeño estudio del cumplimiento que ofrecen las diferentes soluciones utilizadas que puede ser consultado en el anexo A. La herramienta utilizada finalmente ha sido la modificación del kernel linux RTAI que proporciona tiempo real. Se ha realizado una librería de utilización cuyas funciones pueden ser consultadas en el anexo C.

4.4. Librerías a desarrollar

Para cumplir requisitos del TFG se ha realizado un análisis para determinar cuáles son las funcionalidades que pueden ser encapsuladas en librerías para una vez realizadas y probadas poder ser utilizadas de manera transparente. Puede consultarse más información en el Anexo C.

- **Comedi Manager:** Encapsula el establecimiento de conexión a la tarjeta DAQ, la recepción y la emisión de datos. Presenta ventajas como no necesitar manejar parámetros como el canal de la tarjeta a conectar ya que esa información es almacenada por la librería.
- **Realtime Manager:** Permite simplificar el uso de RTAI ya que el usuario no debe preocuparse por preparar el proceso para convertirlo en un proceso tiempo real.
- **Timeval Manager:** Librería para el manejo de la estructura timeval. Permite realizar operaciones sobre estas estructuras como sumarlas, copiarlas u obtener su representación como cadena de texto. Permite realizar llamadas a nanosleep, que requiere el uso de otra estructura diferente a timeval, de manera transparente.
- **Models Tools:** Aquí se han agrupado herramientas universales para la realización de proyecto como el cálculo de la escala y offset para obtener señales en un mismo rango y otras funciones que realizan análisis de los modelos neuronales, como puede ser el calculo de los puntos producidos por ráfaga.

5

Experimentos realizados y resultados

En este capítulo se presentan los resultados de diferentes experimentos y validaciones llevadas a cabo.

5.1. Requisitos estrictos de temporalidad

Los detalles sobre los métodos empleados en primera instancia y sus resultados pueden ser consultado en el Anexo A. Como conclusión cabe decir que fueron detectados problemas con la utilización de herramientas básicas de linux, subsanadas por medio del uso de software de tiempo real.

En la imagen 5.1 se puede observar los problemas ocasionados por la no utilización de software de tiempo real a pesar de contar con las máximas optimizaciones posibles. En cambio en la imagen de la tabla 3.1 se puede observar una captura similar sin problemas gracias al uso de RTAI.

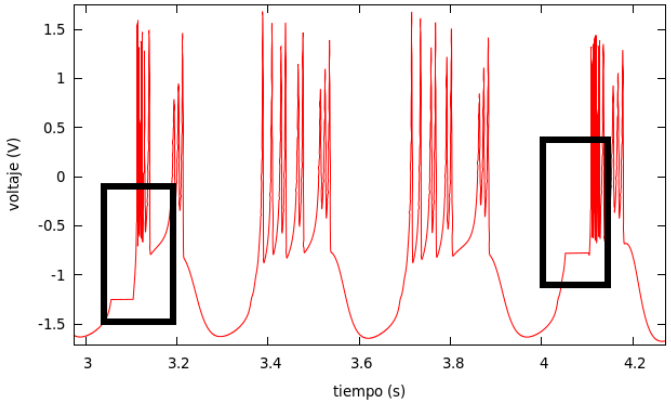


Figura 5.1: Captura de la señal de la neurona electrónica por medio de la tarjeta DAQ sin el uso de RTAI en la cual se señalan zonas donde se observan errores en la recogida de datos.

5.2. Librerías

Para la validación de las librerías se han llevado a cabo diferentes pruebas, dependiendo de las características de cada una. Si se desea obtener más información sobre las librerías véase el capítulo 4.4 y el Anexo B.

- **Comedi Manager:** Se han realizado pruebas enviando y recibiendo datos. Para el primer caso se ha comprobado que los resultados enviados a través de la tarjeta DAQ y visualizados en un osciloscopio correspondían al modelo que se estaba ejecutando como prueba. Para el caso de la recepción se ha comprobado que la recepción de la señal emitida por la neurona electrónica correspondía a la observada en un osciloscopio.
- **Timeval Manager:** Se han realizado test de caja blanca y caja negra para comprobar que los calculos de las estructuras son correctos. Para observar las limitaciones de la función nanosleep se han realizado pruebas detalladas en el Anexo B.
- **Realtime Manager:** Por medio de la librería Timeval Manager se han obtenido tiempos para comprobar si los tiempos de respuesta requeridos eran cumplidos con esta tecnología.
- **Models Tools:** La información de los modelos ha sido validada visualmente en las gráficas de los modelos. Los resultados de la calibración también han sido comprobados por medio de la observación de los resultados, primeramente sobre parejas de modelos y a continuación con la neurona electrónica y un circuito híbrido.

  Todas estas pruebas han sido satisfactorias.

5.3. Interacción bidireccional entre neuronas software

Sin necesidad de contar con una tarjeta DAQ o utilizar las librerías COMEDI ya es posible realizar las primeras pruebas a modo de validación. Para estas pruebas en vez de contar con una conexión a una neurona electrónica o viva contamos con un segundo modelo cuya señal es utilizada cómo si fuera la señal recibida, y a la cual inyectamos los valores que sean enviados desde nuestro modelo a la conexión.

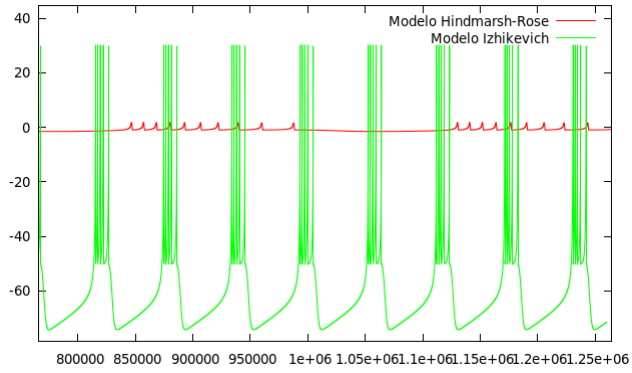
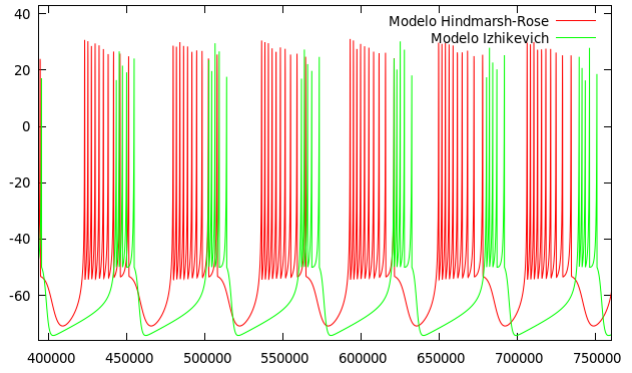
En primer lugar se realizan los cálculos necesarios para autocalibrar el escalado de las señales entre modelos y se definen los puntos que produce cada modelo por ráfaga para definir aproximadamente los puntos a saltar por uno de los modelos. El resultado de este proceso puede observarse en la tabla 5.2.

Los resultados obtenidos para esta autocalibración de la escala de los modelos Hindmarsh-Rose e Izhikevich pueden observarse en la tabla 5.1. Puede comprobarse la validez de los datos obtenidos automáticamente observando la gráfica de los modelos sin escalar de la tabla 5.2. Aplicando los algoritmos diseñados (véase el punto 4.2) se consigue el resultado satisfactorio observado en la segunda imagen de la citada tabla.

Una vez conseguida la calibración de escala se inicia la conexión de los modelos, buscando la conductancia adecuada para conseguir una conexión activa. Para ello se han aplicado los dos métodos establecidos durante la fase de diseño, siendo necesario en este paso establecer un umbral del offset considerado como suficiente para considerar las señales sincronizadas para el método de offset y frecuencia y un umbral de diferencias para el método de la diferencia de potenciales.

Valor máximo del potencial de Hindmarsh-Rose	1.797032
Valor mínimo del potencial de Hindmarsh-Rose	-1.608734
Valor máximo del potencial de Izhikevich	30.240260
Valor mínimo del potencial de Izhikevich	-74.235098
Escala Hindmarsh-Rose hacia Izhikevich	30.676023
Offset Hindmarsh-Rose hacia Izhikevich	-24.885536
Escala Izhikevich hacia Hindmarsh-Rose	0.032599
Offset Izhikevich hacia Hindmarsh-Rose	0.811237

Cuadro 5.1: Valores obtenidos de las lecturas del modelo Hindmarsh-Rose y el modelo Izhikevich y valores determinados para escalar cada uno de los modelos al otro modelo.

Modelos diferentes sin escalado	
Modelos diferentes con escalado	

Cuadro 5.2: Lectura inicial del modelo Hindmarsh-Rose y el modelo Izhikevich en la cual se observar las diferencias de escala y frecuencia y resultado del proceso de escalado, mostrando el modelo Hindmarsh-Rose escalado a las unidades del modelo Izhikevich y con la frecuencia ajustada.

Para el primero de los métodos fue determinada una diferencia entre los offset de ambas señales de 5.000 puntos con una ventana de dos segundos. Disminuir este umbral proporcionara unas conductancias mayores, consiguiendo una sincronización mayor. Las pruebas determinaron que con conseguir un offset bueno podemos ignorar la frecuencia, pues al conseguir determinar un componente el otro queda también fijado correctamente. En la imagen 5.2 se observa como se ajusta la conductancia de dos modelos.

Para el segundo método se obtuvieron similares resultados. De estas pruebas se determinó que el uso de las frecuencias de disparo no era necesario, aunque también puede ser utilizado. Además se observa cómo el criterio de diferencias entre los potenciales puede variar mucho su umbral adecuado según las posibles formas que puedan tener las diferentes neuronas y modelos, siendo un valor más robusto en el caso del umbral del offset.

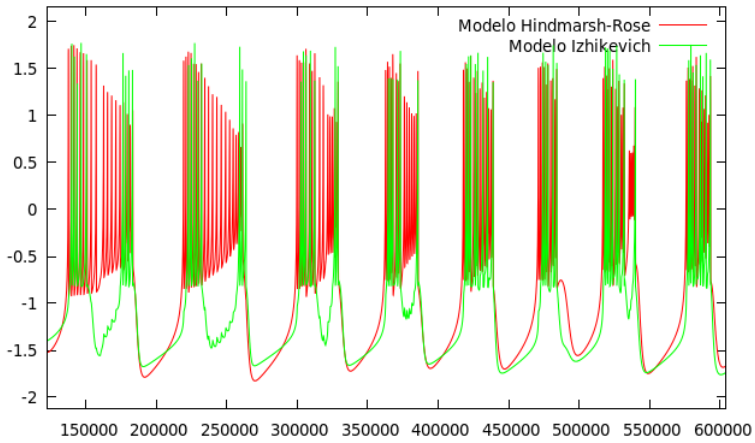


Figura 5.2: Proceso de sincronización entre los modelos Hindmarsh-Rose y Izhikevich por el método de diferencia de offset y frecuencia entre modelos. El proceso dura cuatros ventanas de observación, consistiendo cada una en dos ráfagas.

5.4. Interacción bidireccional entre neuronas software y neuronas electrónicas

Una vez comprobado el funcionamiento de los algoritmos el siguiente paso que se realizó fue la eliminación de segundo modelo, sustituyéndose por la recepción y envío de datos a la tarjeta DAQ, comenzando a utilizar COMEDI. En un primer momento fue conectada la neurona electrónica del GNB (véase 2.3.1), que implementa el modelo HindmarshRose. Los resultados obtenidos por lo tanto fueron similares a los obtenidos utilizando dos modelos, permitiendo esto validar la correcta utilización de COMEDI y RTAI. En la figura 5.3 puede observarse una sincronización entre el mapa de Rulkov y la neurona electrónica del GNB.

Adicionalmente se observó que el ajuste de la ráfaga para que su duración sea la misma que la de la ráfaga leída de la tarjeta podría servir para poder prescindir de la fijación del valor de la conductancia, escogiendo un valor mínimo y fijo. Esto es debido a que cuando las dos señales tiene una escala igual y sus ráfagas tienen igual frecuencia solo es necesario una conexión mínima entre los modelos para que sincronicen su momento de disparo.

La realización de este cálculo para conseguir ráfagas de igual duración conllevó también el dejar de utilizar la frecuencia como método efectivo de selección de la conductancia, debido a que muestra y modelo presentan igual frecuencia, debiendo ser ajustado solo el offset.

Para comprobar la sincronización podemos comparar los niveles de ambos potenciales. La figura 5.4 muestra como los datos no tienen relación cuando no existe sincronización, sin embargo cuando el nivel de sincronización es adecuado los potenciales establecen una relación que se ajusta a la recta x=y, como se puede observar en la figura 5.5. Otra alternativa es calcular el error cuadrático medio, y comprobar como este valor se reduce (ver figuras 5.6 y 5.7).

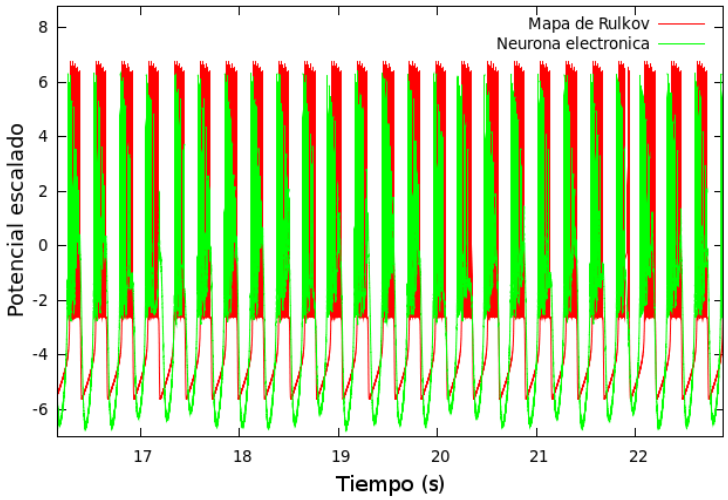


Figura 5.3: Sincronización entre el mapa de Rulkov y la neurona electrónica del GNB que implementa el modelo HindmarshRose.

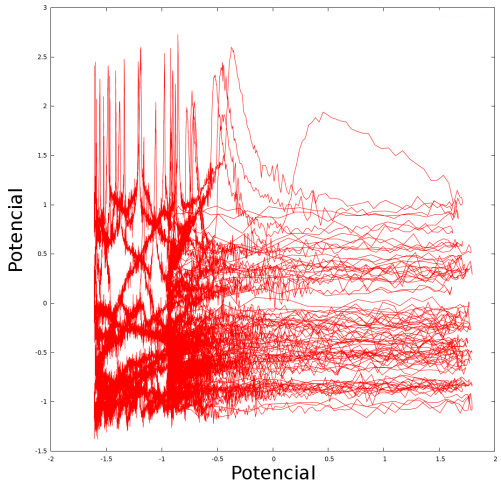


Figura 5.4: Relación de los potenciales entre dos neuronas no sincronizadas.

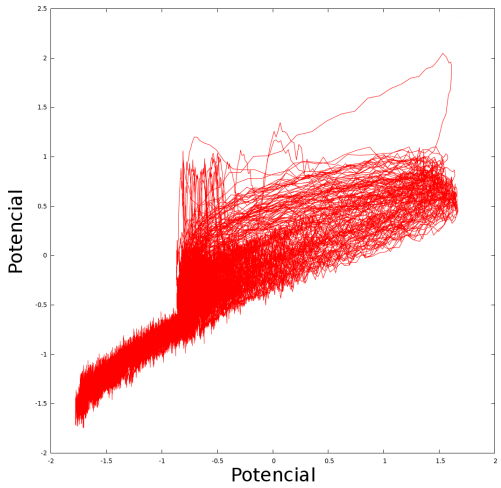


Figura 5.5: Relación de los potenciales entre dos neuronas sincronizadas.

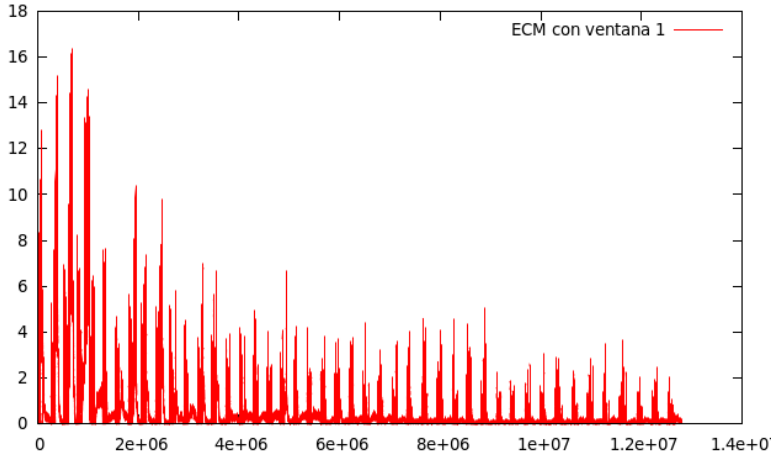


Figura 5.6: ECM de la sincronización entre dos neuronas con una ventana de observación para el cálculo de 1 punto.

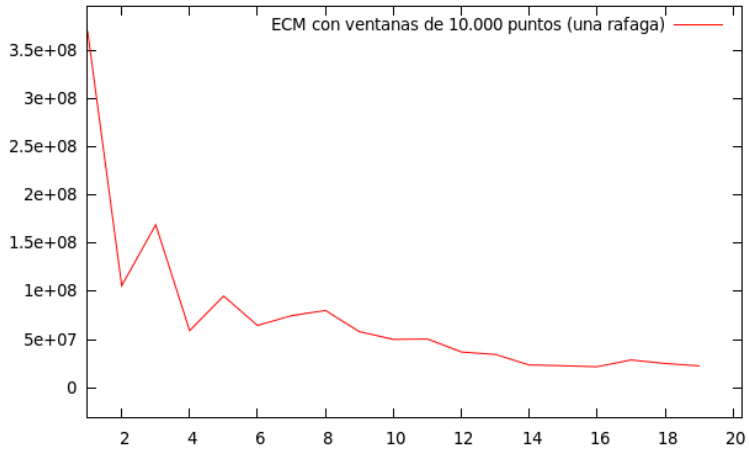


Figura 5.7: ECM de la sincronización entre dos neuronas con una ventana de observación para el cálculo de 10.000 puntos.

5.5. Circuitos híbridos

Una vez comprobado el correcto funcionamiento de los algoritmos fueron validados en circuitos híbridos (véase 2.5), utilizando para ello CPG (véase 2.6) extraídos del cangrejo de mar común.

El escalado de las señales demostró funcionar de manera robusta y de acuerdo al comportamiento esperado. El método de selección de conductancia por offset y frecuencia también funcionó, sin embargo el ruido puede afectar a las medidas. Para el método de selección de conductancia por diferencia de potenciales se obtuvieron resultados correctos, aunque se demostró positivo incrementar la ventana de observación y fue necesario cambiar el umbral, ambos cambios propiciados por los niveles de ruido con los que cuenta la señal biológica respecto a la señal producida por un modelo neuronal.

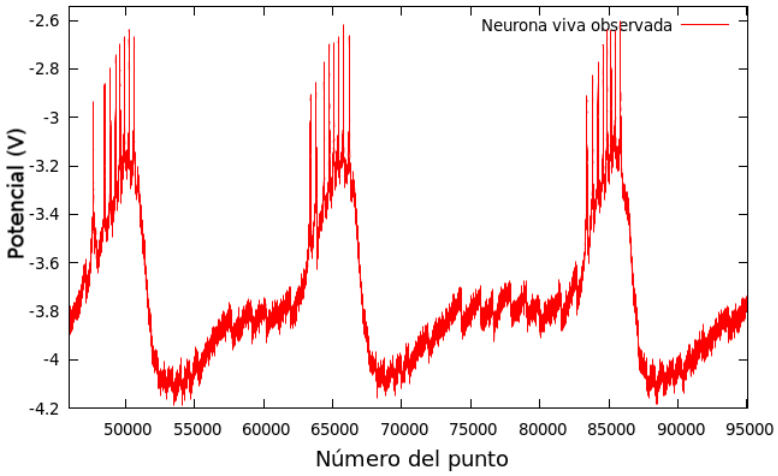


Figura 5.8: Señal obtenida de la neurona real utilizada para realizar los experimentos. Puede observarse la gran cantidad de ruido que presentan estos datos frente a los modelos.

5.5.1. Selección de conductancia por offset y frecuencia

En la figura 5.9 puede observarse como progresivamente las ráfagas comienzan a producirse en el mismo instante, hasta que finalmente se pueden ver tres ráfagas simultáneas en ambas señales. También se puede observar cómo las neuronas reales muestras unos comportamientos no normales al variar la conductancia, observando en el segundo 8 como por ejemplo que el disparo no se produce adecuadamente, solucionándose el problema cuando se interrumpe la variación de la conductancia, como puede observarse en la figura 5.10.

Los valores de conductancia calculados fueron de 0.15.

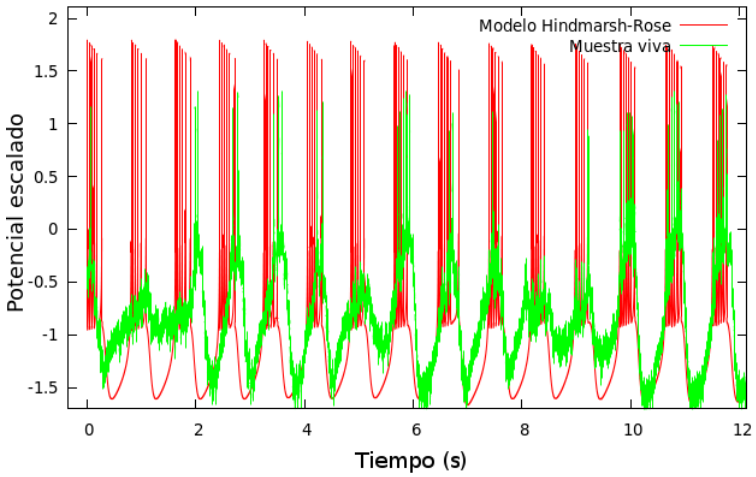


Figura 5.9: Proceso de sincronización y ajuste del valor de conductancia por medio del método de offset y frecuencia con una neurona real.

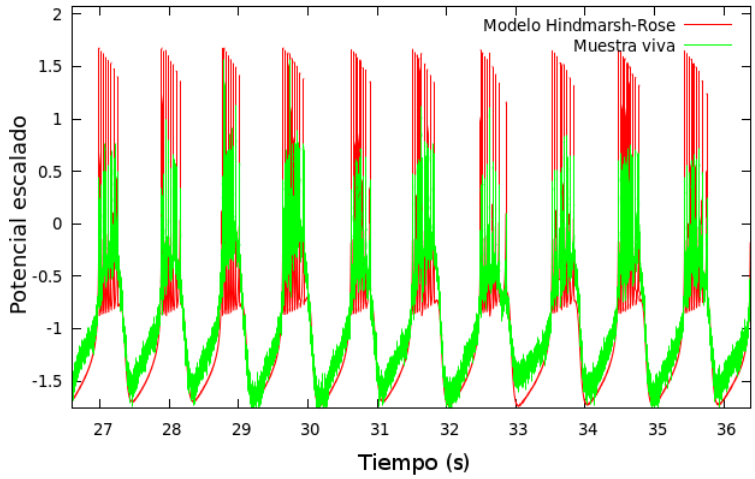


Figura 5.10: Neurona viva sincronizada con un modelo neuronal.

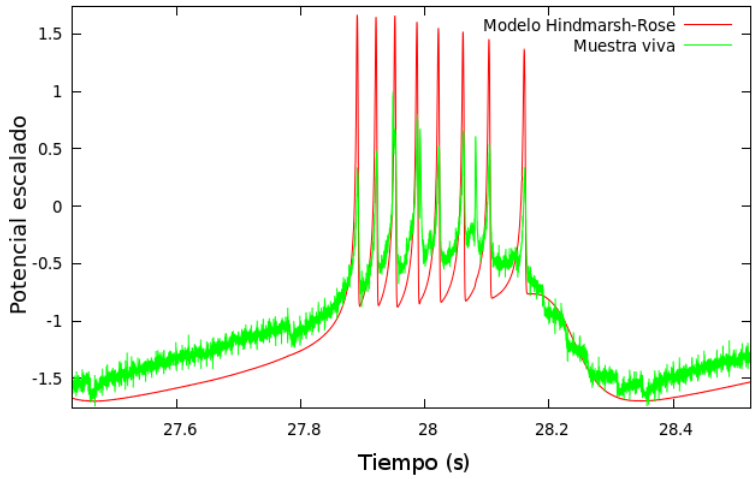


Figura 5.11: Detalle de una ráfaga sincronizada.

5.5.2. Selección de conductancia por diferencia de potenciales

Los valores obtenidos por medio de este método fueron similares, siendo ligeramente inferiores. Este comportamiento puede ser ajustado por medio de los umbrales. Por otro lado esto se refleja en los resultados, donde puede ser observada una sincronización menos pasiva (figura 5.12) si observamos la zona de de reposo y la fase refractaria.

5.5.3. Conexión en antifase

Si invertimos el signo de las señales que se envían y se introducen en el modelo podemos realizar una conexión en antifase. La figura 5.13 muestra la conexión establecida durante la experimentación. La conductancia fue fijada en 0.03. Este tipo de conectividad simula sinapsis inhibidora entre neuronas.

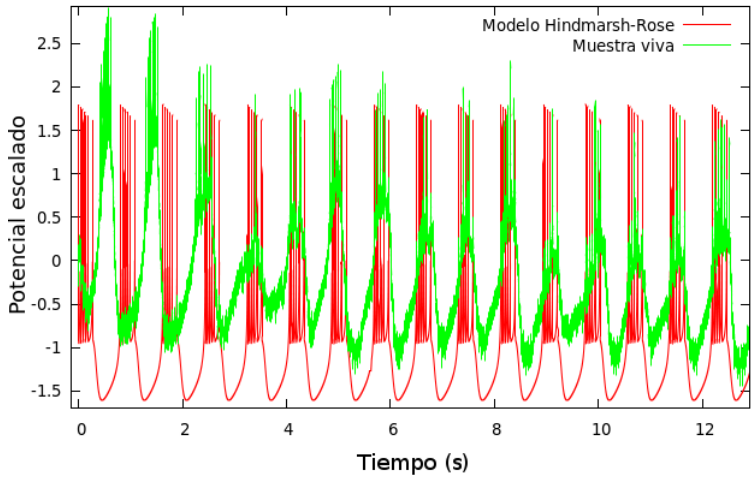


Figura 5.12: Proceso de sincronización y ajuste del valor de conductancia por medio del método de diferencia de potenciales con una neurona real.

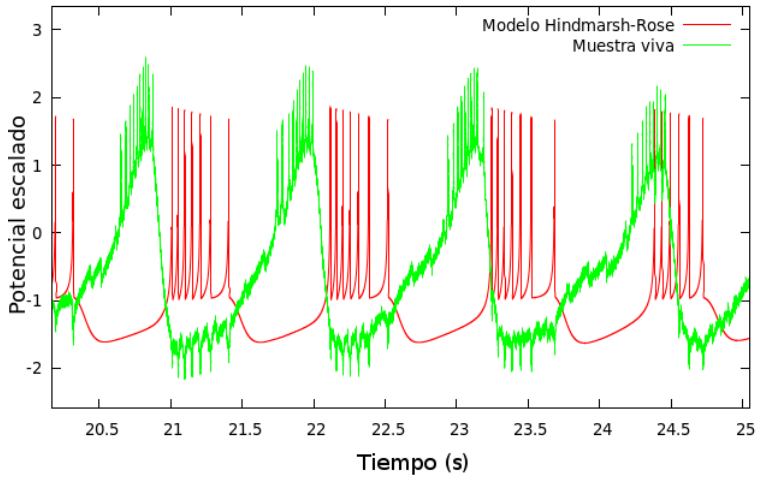


Figura 5.13: Conexión con una sincronización antifase.

5.5.4. Regularización de ritmos neuronales

La regularidad de los ritmos neuronales de los CPG puede verse afectada si modificamos el medio de la preparación por medio de etanol. Tras esto podemos utilizar nuestro modelo neuronal para tratar de regularizar estos ritmos devolviéndolos a la normalidad. Replicar estos comportamientos y tratar de revertirlos es de especial interés en el contexto del control de la actividad neuronal patológica (ver sección 2.7). En la figura 5.14 puede observarse el comportamiento caótico que presentaba la neurona después de ser afectada la preparación por el etanol. En la figura 5.15 se observa cómo el modelo consigue imponer la realización de un ritmo constante.

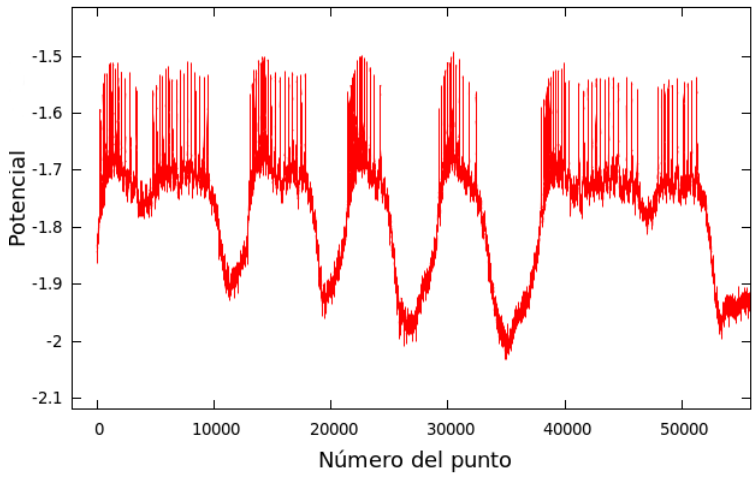


Figura 5.14: Actividad neuronal irregular por el efecto del etanol. Se observa variación en la duración de las ráfagas.

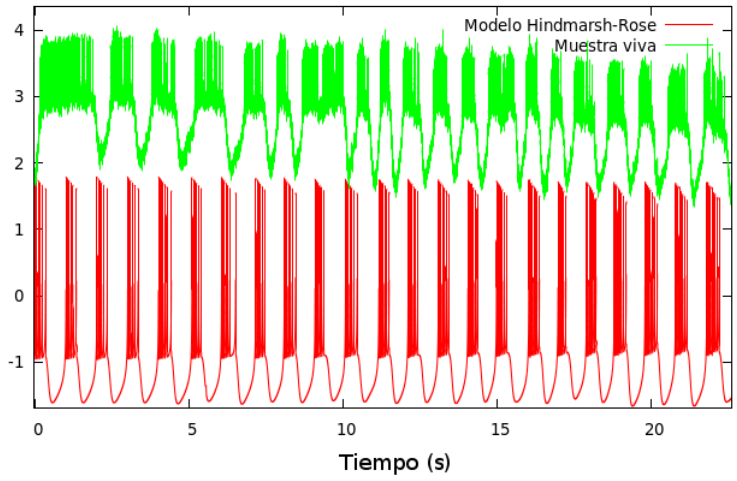


Figura 5.15: Evolución de una neurona con actividad irregular forzada por el modelo neuronal realizado a realizar progresivamente ritmos regulares. Durante la segunda mitad temporal de la imagen el valor de conductancia ha quedado fijado, estableciéndose una frecuencia y tiempo de disparos regulares.

6

Conclusiones y trabajo futuro

6.1. Conclusiones

Se han cumplido los objetivos marcados al inicio del proyecto y su objetico principal, establecer un mecanismo de autocalibración de los parámetros que afectan a la conexión de un modelo neuronal cuando se realizan circuitos híbridos. Para ello el proyecto ha contado con varias etapas que se han resuelto satisfactoriamente. En una primera etapa se ha conseguido:

- Implementar modelos neuronales con comportamientos en ráfagas.
- Incorporar tecnología de tiempo real para cumplir los requisitos de temporalidad de los circuitos híbridos y enviar y recibir información satisfactoriamente a la tarjeta DAQ.

Posteriormente se han diseñado y utilizado los algoritmos de autocalibración que han sido validados y se ha demostrado su funcionamiento mediante su utilización en un circuito híbrido.

Por un lado el algoritmo de escalado de la señal es un método universal entre modelos y experimentos. Por otro lado el algoritmo que determina el valor de conductancia funciona correctamente pero cuenta con parámetros que pueden ser modificados, determinando un valor de conductancia u otro. En ambos casos hay que tratar con el ruido de las señales biológicas para determinar los valores umbrales adecuados para un nivel de sincronización adecuado. Finalmente el tiempo de ráfaga de los modelos también se adapta a lo observado en la neurona real.

Se ha demostrado la necesidad de la utilización de la autocalibración cuando se realicen circuitos híbridos debido al ahorro de tiempo que ha supuesto en la experimentación respecto a una búsqueda manual de los parámetros. Por otro lado en los experimentos de validación se ha podido regularizar un circuito CPG que producía ritmos no regulares por el etanol.

Adicionalmente se han elaborado librerías de los desarrollos más genéricos realizados y de los circuitos realizados y estudiado soluciones para cumplir los requisitos de estricta temporalidad si no se cuenta con un software de tiempo real.

6.2. Trabajo futuro

La posibilidad de autocalibrar circuitos híbridos puede permitir la generalización de este tipo de tecnología en los estudios de electrofisiología neuronal. Este trabajo puede tener continuación de las siguientes maneras:

- Incorporación de los algoritmos de autocalibración a la plataforma RTBiomanager del GNB para implementar y visualizar en tiempo real circuitos híbridos.
- Refinamiento de la búsqueda del valor de conductancia adecuado con métodos más complejos de análisis de la señal o mediante técnicas de filtrado y detección de eventos de la señal biológica obtenida.
- Validación de la búsqueda de conductancia en modelos neuronales de conductancia más complejos.
- Estudio de los valores de conductancia necesarios en diferentes tipos de neuronas correspondiente a distintos circuitos neuronas, pudiendo así demostrar la rápida adaptación a cada tipo de dinámica neuronal.
- Incorporar algoritmos de detección de eventos en las señales en tiempo real para modular la interacción o el estímulo en circuitos híbridos.
- Publicación del algoritmo en una revista científica y difusión en congresos internacionales.

Bibliografía

Benabid, A. L., Chabardes, S., Mitrofanis, J., and Pollak, P. (2009). Deep brain stimulation of the subthalamic nucleus for the treatment of parkinson's disease. *Lancet Neurol*, 8:67–81.

Berenyi, A., Belluscio, M., Mao, D., and Buzsaki, G. (2012). Closed-loop control of epilepsy by transcranial electrical stimulation. *Science*, 337:735–737.

Chamorro, P., Muñiz, C., Levi, R., Arroyo, D., Rodríguez, F., and Varona, P. (2012). Generalization of the dynamic clamp concept in neurophysiology and behavior. *PLoS ONE*, 7.

Chen, W. R., Midtgaard, J., and Shepherd, G. M. (1997). Forward and backward propagation of dendritic impulses and their synaptic control in mitral cells. *Science*, 278:463–467.

Cohen, A., Shappir, J., Yitzchaik, S., and Spira, M. E. (2006). Experimental and theoretical analysis of neuron-transistor hybrid electrical coupling: The relationships between the electro-anatomy of cultured aplysia neurons and the recorded field potentials. *Biosensors and Bioelectronics*, 22:656–663.

Cole, K. (1968). *Membranes, Ions and Impulses: A Chapter of Classical Biophysics.* Univ. Calif. Press. Berkeley.

Cole, K. S. and Curtis, H. J. (1939). Electric impedance of the squid giant axon during activity. *The Journal of general physiology*, 22:649–670.

Connor, J. A., Walter, D., and McKown, R. (1977). Neural repetitive firing: modifications of the hodgkin-huxley axon suggested by experimental results from crustacean axons. *Biophysical journal*, 18:81–102.

Destexhe, A. and Bal, T. (2009). *Dynamic Clamp: From Principles to Applications.* Springer.

Elson, R. C., Selverston, A. I., Huerta, R., Rulkov, N. F., Rabinovich, M. I., and Abarbanel, H. D. I. (1998). Synchronous behavior of two coupled biological neurons. *Phys. Rev. Lett.*, 81:5692–5695.

Fernandez−Vargas, J., Pfaff, H. U., Rodríguez, F. B., and Varona, P. (2013). Assisted closed-loop optimization of ssvep-bci efficiency. *Frontiers in neural circuits*, 7.

Gómez Nicola, D. and Nieto Sampedro, M. (2008). Glía reactiva. *Mente y Cerebro*, 32:78–87.

Harris-Warrick, R. M. (2011). Neuromodulation and flexibility in central pattern generator networks. *Current Opinion in Neurobiology*, 21:685–692.

Hebb, D. O. (1949). *The Organization of Behaviour.* Organization.

Herrero-Carrón, F., Rodríguez, F. B., and Varona, P. (2011). Bio-inspired design strategies for central pattern generator control in modular robotics. *Bioinspiration & biomimetics*, 6.

Hille, B. (1977). Ionic channels in excitable membranes. *Biophysical Journal*, 22:283–294.

Hindmarsh, J. and Rose, R. (1984). A model of neuronal bursting using three coupled first order differential equations. *Proceedings of the Royal Society of London. Series B, Biological sciences*, 221:87–102.

Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its applicaiton to conduction and excitation in nerve. *J Physiol*, 117:500–544.

Hooper, S. (2001). Central pattern generators. *eLS*, 221:1–12.

Hormuzdi, S. G., Filippov, M. A., Mitropoulou, G., Monyer, H., and Bruzzone, R. (2004). Electrical synapses: a dynamic signaling system that shapes the activity of neuronal networks. *Biochimica et Biophysica Acta (BBA) - Biomembranes*, 1662:113137.

Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14:1569–1572.

Kandel, E. R., Schwartz, J. H., and Jessell, T. M. (2000). Principles of neural science. *Neurology*, 4.

Kepler, T. B., Abbott, L. F., and Marder, E. (1992). Reduction of conductance-based neuron models. *Biological Cybernetics*, 66:381–387.

Le Masson, G., Renaud-Le Masson, S., Debay, D., and Bal, T. (2002). Feedback inhibition controls spike transfer in hybrid thalamic circuits. *Nature*, 417:854–858.

López-Muñoz, F., Boya, J., and Alamo, C. (2006). Neuron theory, the cornerstone of neuroscience, on the centenary of the nobel prize award to santiago ramón y cajal. *Brain Res Bull*, 70:391405.

M. Hooper, R. M., Tikidji-Hamburyan, R. A., Canavier, C. C., and Prinz, A. A. (2015). Feedback control of variability in the cycle period of a central pattern generator. *Journal of Neurophysiology*, 114:2741–2752.

Manor, Y. and Nadim, F. (2001). Frequency regulation demonstrated by coupling a model and a biological neuron. *Neurocomputing*, 38-40:269–278.

Mcculloch, W. S. and Pitts, W. (1943). A logical calculus of the idea immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133.

Morris, C. and Lecar, H. (1981). Voltage oscillations in the barnacle giant muscle fiber. *Biophysical journal*, 35:193–213.

Nagumo, J., Arimoto, S., and Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50:2061–2070.

Nowotny, T., Szucs, A., Pinto, R. D., and Selverston, A. I. (2006). Stdpc: A modern dynamic clamp. *Journal of Neuroscience Methods*, 158:287–299.

Pinto, R. D., Varona, P., Volkovskii, A. R., Szücs, A., Abarbanel, H. D. I., and Rabinovich, M. I. (2000). Synchronous behavior of two coupled electronic neurons. *Physical Review E*, 62.

Potter, S. M., El Hady, A., and Fetz, E. E. (2014). Closedloop neuroscience and neuroengineering. *Frontiers in neural circuits*, 8.

Prescott, S. A., De Koninck, Y., and Sejnowski, T. J. (2008). Biophysical basis for three distinct dynamical mechanisms of action potential initiation. *PLoS Computational Biology*, 4.

Rabinovich, M. I., Varona, P., Selverston, A. I., and Abarbanel, H. D. I. (2006). Dynamical principles in neuroscience. *Reviews of Modern Physics*, 78.

Ramón y Cajal, S. (1909). *Histologie du Systeme Nerveux de lHomme et des Vertebres.* Paris: Maloine.

Roth, E., Sponberg, S., and Cowan, N. J. (2014). A comparative approach to closed-loop computation. *Current Opinion in Neurobiology*, 25:54–62.

Rulkov, N. F. (2002). Modeling of spiking-bursting neural behavior using two-dimensional map. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 65.

Sabbatini, R. (2003). Neurons and synapses: The history of its discovery. *Brain & Mind Magazine*, 17.

Selverston, A. I., Rabinovich, M. I., Abarbanel, H. D. I., Elson, R., Szücs, A., Pinto, R. D., Huerta, R., and Varona, P. (2000). Reliable circuits from irregular neurons: A dynamical approach to understanding central pattern generators. *Journal of Physiology-Paris*, 94:357–374.

Skinner, F. K., Turrigiano, G. G., and Marder, E. (1993). Frequency and burst duration in oscillating neurons and two-cell networks. *Biological Cybernetics*, 69:375–383.

Stigen, T., Danzl, P., Moehlis, J., and Netoff, T. (2011). Controlling spike timing and synchrony in oscillatory neurons. *Journal of neurophysiology*, 105:2074–2082.

Szücs, A., Varona, P., Volkovskii, A. R., Abarbanel, H. D., Rabinovich, M. I., and Selverston, A. I. (2000). Interacting biological and electronic neurons generate realistic oscillatory rhythms. *Neuroreport*, 11:563–569.

Van Boxtel, G. J. and Gruzelier, J. H. (2014). Neurofeedback: introduction to the special issue. *Biol Psychol*, 95:1–3.

Varona, P., Torres, J. J., Abarbanel, H. D. I., Rabinovich, M. I., and Elson, R. C. (2001). Dynamics of two electrically coupled chaotic neurons: Experimental observations and model analysis. *Biological Cybernetics*, 84:91–101.

Williams, R. W. and Herrup, K. (1988). The control of neuron number. *Annual review of neuroscience*, 11:423–453.

Yarom, Y. (1991). Rhythmogenesis in a hybrid system-interconnecting an olivary neuron to an analog network of coupled oscillators. *Neuroscience*, 44:263–275.

Anexo A: Cumplimiento de los requisitos de temporalidad

Soluciones básicas

Cpuset

Cpuset es una interfaz para controlar la ubicación de ejecución y de memoria de un programa.

Se puede consultar más información en la siguiente dirección `http://manpages.ubuntu.com/manpages/xenial/en/man7/cpuset.7.html`.

El kernel debe compilarse con este sistema, pero suele estar incluido en la mayoría de distribuciones típicas de linux.

Para ser utilizado podemos hacer uso de una herramienta en Python que nos permite utilizar estas herramientas con facilidad por medio de la linea de comandos. La información sobre como crear grupos de procesadores, eliminar los máximos procesos posibles de un grupo y como ejecutar una aplicación en uno de los grupos puede ser consultada en `https://rt.wiki.kernel.org/index.php/Cpuset_Management_Utility`.

El objetivo de utilizarlo es que nuestro proceso tenga que competir lo mínimo posible por los recursos del procesador, pudiendo ejecutarse en tiempo real. Sin embargo el no contar con un planificador de tiempo real y competir con procesos del sistema con mayor prioridad no movibles de un procesador hace que si bien mejoren los resultados de uso no se alcancen aún resultados óptimos.

Existen también soluciones para arrancar un sistema linux con núcleos reservados que anteriormente conseguían excluir aún más procesos que cpuset, aunque actualmente el efecto conseguido es el mismo.

Nanosleep

El uso de cpuset debe ser combinado con un método de espera. Para ello se utilizó primeramente la función C de linux llamada nanosleep (`http://man7.org/linux/man-pages/man2/nanosleep.2.html`).

Para la utilización de esta función y realizar los cálculos de los tiempos de espera necesarios se ha realizado una biblioteca que puede ser consultada en el Anexo C.

La utilización de nanosleep sin embargo no garantiza que la ejecución se reanude en el tiempo deseado. Para comprobar este problema se realizaron esperas medidas por medio del comando de terminal time (teniendo el programa ejecutado solo una llamada a nanosleep) u obteniendo el tiempo en las instrucciones anterior y posterior de la llamada y calculando la diferencia por medio de las funciones correspondiente de la librería, obteniendo los resultados de la tabla 7.1.

Tiempo deseado (s)	Tiempo obtenido (s)
0.000040	0.000109
0.000040	0.000100
0.000060	0.000120

Cuadro 7.1: Resultados de esperas deseadas y esperas obtenidas con el uso de nanosleep

Como se puede observar en la tabla la precisión de nanosleep en un sistema típico de uso general es insuficiente para el proyecto llevado a cabo, por lo cual el uso de esta función fué sustituido por una espera activa, observando continuamente el tiempo actual hasta llegar al momento deseado, solución con la cual se obtenían distancias menores entre los valores deseado y obtenido.

Más optimizaciones

Para una mayor prioridad de nuestro proceso podemos ejecutar el programa precedido por el comando "nice -n<PRIORIDAD> ./Programa" pudiendo establecer una prioridad desde -20 a 19. Siendo la mayor prioridad -20 y la mínima 19.

También odemos establecer la política del planificador de procesos a una más beneficiosa para nuestro programa:

```
#/bin/bash
struct sched_param prior;
prior.sched_priority=sched_get_priority_max(SCHED_FIFO);
sched_setscheduler(0, SCHED_FIFO, &prior);
```

Por último podemos establecer una prioridad de procesador para las interrupciones, definiendo máscaras para indicar en que núcleos pueden o no ejecutarse. Se puede consultar información sobre esto en `https://cs.uwaterloo.ca/~brecht/servers/apic/SMP-affinity.txt`.

Para realizar los cambios de prioridad se realizó el siguiente script:

```
#/bin/bash

for d in $(find /proc/irq/ -maxdepth 1 -type d)
do
  echo 0f > $d/smp_affinity
  echo $d
done

for d in $(find /proc/irq/ -maxdepth 1 -type d)
do
  cat $d/smp_affinity_list
done
```

El primer punto a valorar es analizar si se consigue calcular los puntos antes de que deban ser enviados. Este requisito fue cumplido incluso sin ningún método especial, ya que los modelos elegidos son rápidos y la potencia computacional actual es suficiente.

El segundo punto a valorar es conseguir que el envío de un punto a la tarjeta sea producido cada 1/frecuencia de la tarjeta segundos. Considerando una frecuencia de 10kHz y utilizando las mejores soluciones (esto es cpuset, espera activa, nice y movimiento de interrupciones) se consiguieron resultados como el observado en la imagen 7.2. Sin embargo el resultado obtenido no era siempre bueno, pues a pesar de conseguir reducir su impacto un proceso del kernel o una interrupción podían afectar a la ejecución, como se observa en la imagen 7.3.

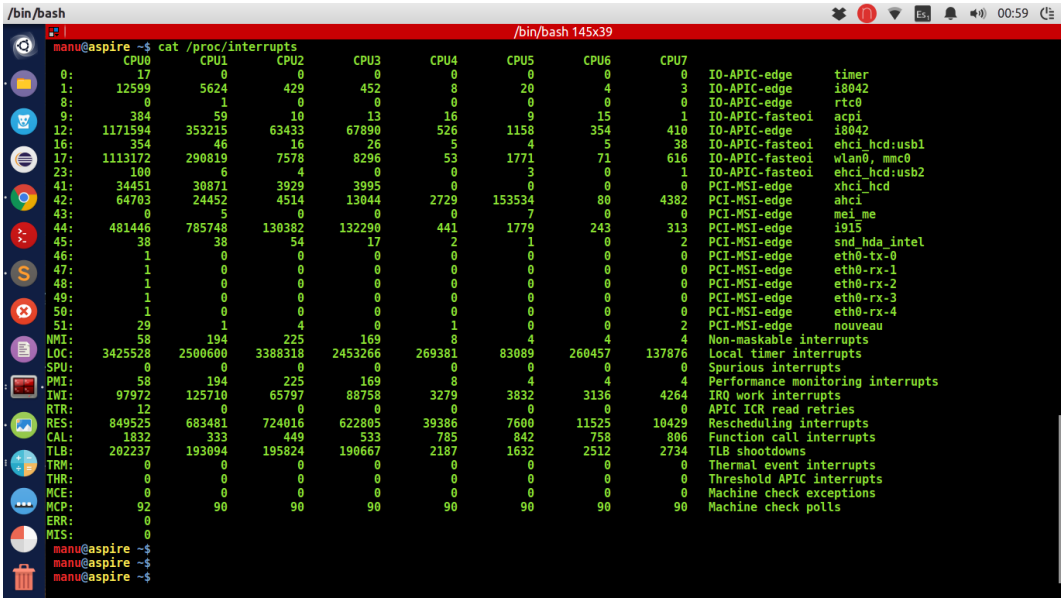


Figura 7.1: Imagen en la cual se puede observar como los núcleos del 4 al 7 cuentan con un menor número de interrupciones del sistema atendidas.

Resultados

Algunas estadísticas fruto del análisis de los datos obtenidos (mediante ráfagas de envío de información de un minuto) indicaron que el 98 % de los datos eran enviados en tiempo, siendo el retraso del 2 % restante valores inferiores al momento de envío del punto posterior en el 90 % de los casos. Los puntos restantes, si bien siendo una cantidad mínima ocasionaban una secuencia de entre 10 y 12 puntos enviados incorrectamente.

Esta cantidad de puntos perdidos puede afectar a la conexión, quizás no lo suficiente como para no sea usable, pero contando con una solución como RTAI, que no requiere la necesidad de preparar y utilizar múltiples soluciones y proporciona un resultado mejor llevaron a no utilizar estos métodos durante las validaciones finales.

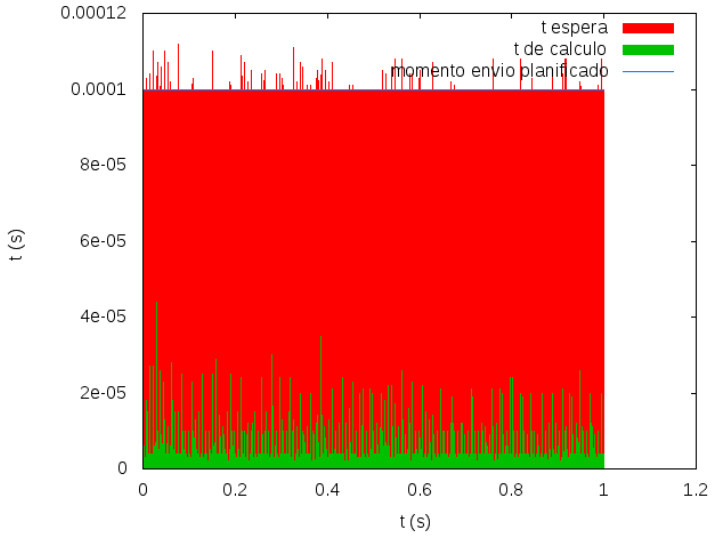


Figura 7.2: Ejemplo positivo de tiempo de cálculo de puntos y desvío del momento de envío con soluciones no tiempo real.

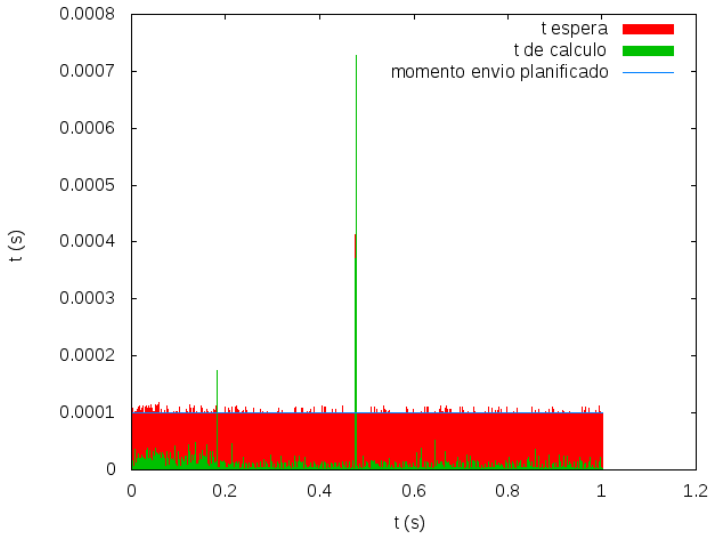


Figura 7.3: Ejemplo de un problema en los tiempos de envío utilizando soluciones no tiempo real.

RTAI

Finalmente se ha utilizado RTAI, realizando una biblioteca (véase Anexo C) para su utilización. La instalación de RTAI es más complicada que las soluciones anteriores, pero aporta mejores resultados, y en el caso de este proyecto ya se encontraba instalado en el ordenador del GNB utilizado en los experimentos.

Los resultados normales son similares aunque ligeramente mejores, la diferencia del uso de RTAI es principalmente que el número de puntos enviados en tiempo incorrecto por causa de interrupciones es más de dos veces menor. Además cuando se produce la recuperación es mucho más rápida.

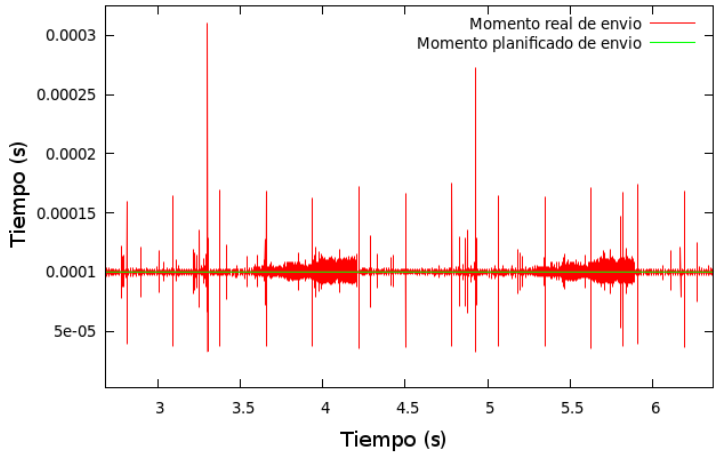


Figura 7.4: Ejemplo del desvío del tiempo de envío respecto al momento planificado usando RTAI.

Anexo B: Detalles de los modelos neuronales

Método de Euler para la resolución de EDOs

El modelo de Hindmarsh-Rose y el modelo de Izhikevich cuentan con ecuaciones diferenciales ordinarias, las cuales deben ser aproximadas mediante algún método matemático. En este proyecto ha sido usado el método de Euler.

En primer lugar podemos expresar una ecuación diferencial como un límite de la siguiente forma:

$$\frac{df(t)}{dt} = \lim_{\triangle t \to 0} \frac{f(t+\triangle t) - f(t)}{\triangle t}$$

Si el incremento de tiempo es muy próximo a cero podemos considerar el resultado como:

$$\frac{df(t)}{dt} = \frac{f(t+\triangle t) - f(t)}{\triangle t}$$

Ahora solo debemos despejar el valor buscado, obteniendo:

$$f(t + \triangle t) = f(t) + \triangle t \frac{df(t)}{dt}$$

Selección del paso de integración

La utilización del método de Euler requiere por lo tanto la selección de un valor $\triangle t$ llamado paso de integración.

Este valor determina la resolución con la cual es producida la señal, un paso muy pequeño proporciona más puntos por ráfaga y un mayor tiempo de cálculo que un paso mayor. El paso sin embargo tampoco puede ser un valor demasiado grande, ya que entonces la resolución de la señal producida será insuficiente.

Por lo tanto el paso de integración debe ser un valor que permita obtener una correcta resolución de la señal y que además no tarde más de un segundo en producir una ráfaga.

En la figura 7.5 se puede observar como en el caso del modelo HindmarshRose un paso de 0.01 empieza a ofrecer una pérdida de resolución, estando los puntos mas separados entre sí. También se observa como para valores de 0.001 y 0.0001 la resolución es similar estando las gráficas superpuestas. Se selecciona por lo tanto un valor de 0.001 como valor de integración para el modelo de HindmarshRose. Tras esto hay que comprobar que el paso es válido obteniendo el tiempo de cálculo de una ráfaga, que como se puede observar en la misma imagen es inferior a un segundo.

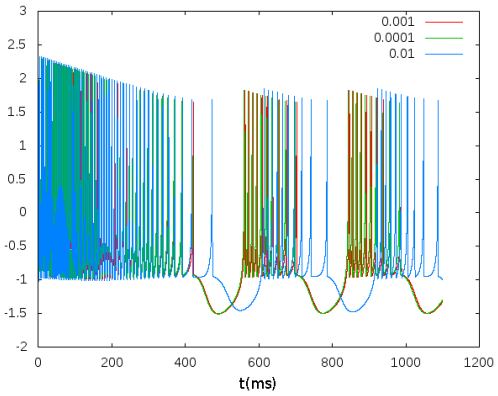


Figura 7.5: Selección del paso de integración para el modelo de Hindmarsh-Rose.

Modelo Hindmarsh-Rose

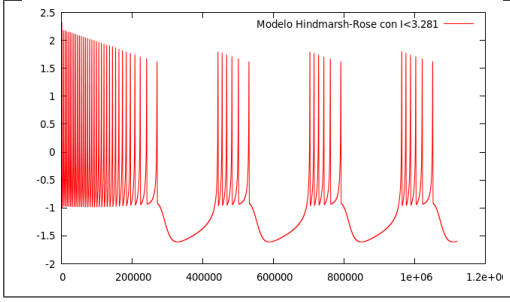
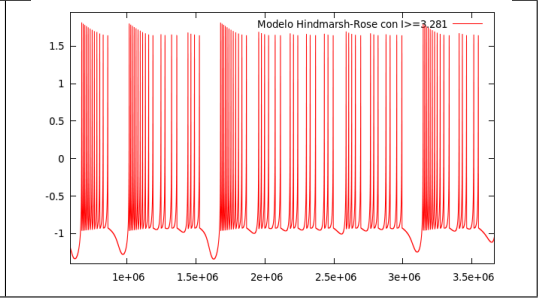
$$\frac{dx(t)}{dt} = ay(t) + bx(t)^2 - cx(t)^3 - dz(t) + I$$
$$\frac{dy(t)}{dt} = e - fx(t)^2 - y(t)$$
$$\frac{dz(t)}{dt} = \mu(-z(t) + S(x(t) + h))$$

Valores iniciales

Parámetro	Valor
a	1.0
b	3.0
c	1.0
z	1.0
I	3.0
e	1.0
f	5.0
μ	0.0021
S	4.0
h	1.6

Cuadro 7.2: Tabla con los valores seleccionados para el modelo Hindmarsh-Rose

Cambios de comportamiento

Modelo actuando de manera regular con I<3.281	Modelo actuando de manera irregular con I>3.281
	

Cuadro 7.3: Modelo de Hindmarsh-Rose con comportamientos regulares o irregulares. Se puede observar también como el modelo converge al inicio de los cálculos con un comportamiento regular.

Mapa de Rulkov

$$x_{n+1} = f(x_n, y_n + eI_n)$$
$$y_{n+1} = y_n - \mu(x_n + 1) + \mu\sigma$$

$$f(x,y) = \begin{cases} \frac{\alpha}{1-x} + y & si \quad x \leq 0 \\ \alpha + y & si \quad 0 < \alpha < \alpha + y \\ -1 & si \quad x \geq \alpha + y \end{cases}$$

Valores iniciales

Parámetro	Valor
α	6.0
μ	0.001
σ	-.01
e	1
I	1

Cuadro 7.4: Tabla con los valores seleccionados para el mapa de Rulkov

Cambios de comportamiento

El mapa de Rulkov puede representar tres tipos de comportamientos. Estos comportamientos se establecen por medio de los parámetros α y μ como puede observarse en la imagen 7.6. La variación de los parámetros dentro de una misma actividad ajusta diferentes valores como la frecuencia de disparo o la longitud de una ráfaga (Rulkov, 2002).

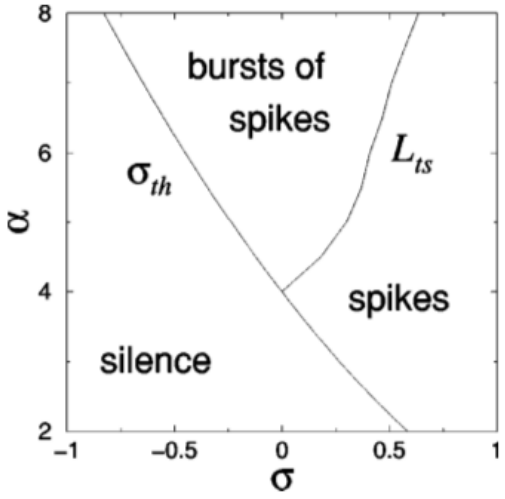
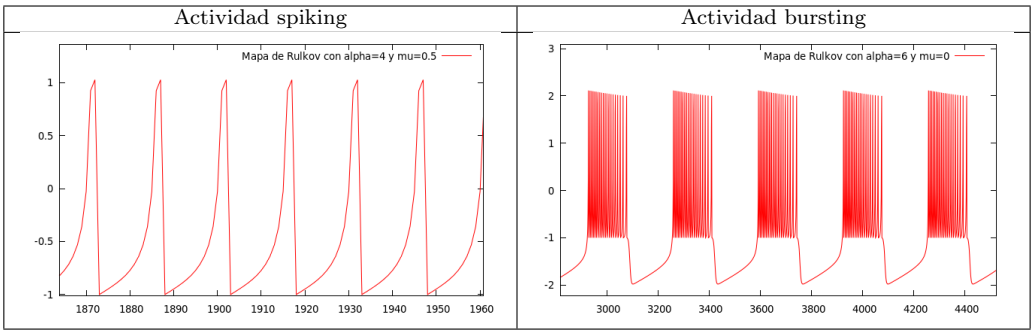


Figura 7.6: Diagrama de bifurcación de la actividad del mapa de Rulkov por el plano (μ, α)



Cuadro 7.5: Mapa de Rulkov generando actividad spiking o actividad bursting dependiendo de los valores μ y α seleccionados.

Interpolación lineal

Este modelo si bien no cuenta con la necesidad de aplicar ningún método de resolución específico produce una cantidad menor de puntos por ráfaga de los necesarios. Para conseguir ráfagas con una resolución de puntos mayor que permita enviar un valor en la frecuencia deseada se ha utilizado un método de interpolación lineal.

Para obtener los puntos necesarios entre dos puntos obtenidos por medio de las fórmulas del modelo se ha utilizado la siguiente función de interpolación:

$$f(x|x_1;x_2) = f(x_1) + \frac{f(x_2)-f(x_1)}{x_2-x_1}(x-x_1)$$

Modelo Izhikevich

$$\frac{dv(t)}{dt} = 0{,}004v^2 + 5v + 140 - u + I$$
$$\frac{du(t)}{dt} = a(bv - u)$$
$$\text{Si } v >= 30 => \begin{cases} v = c \\ u = u + d \end{cases}$$

Valores iniciales

Parámetro	Valor
I	10.0
a	0.02
b	0.2
c	-50.0
d	2.0

Cuadro 7.6: Tabla con los valores seleccionados para el modelo Izhikevich

Cambios de comportamiento

El modelo de Izhikevich permite simular distintos comportamientos neuronales, para ello el propio autor proporciona un programa en el que ajustar los parámetros de modelo y ver los resultados. El programa puede ser descargado en `http://www.izhikevich.org/publications/spikes.htm`.

Anexo C: Librerías desarrolladas

Este anexo incluye la descripción de las funciones desarrolladas para la utilización de diferentes herramientas durante la realización del TFG. Su descripción general puede ser consultada en el punto 4.4.

Las librerías pueden ser consultadas o descargadas para su utilización en la página web `https://github.com/manurs/neuron-models`

Comedi Manager

- **int comedi_iniciar(void)**

  Función que establece la conexión con la tarjeta DAQ. Pueden ser modificadas las variables de programa que seleccionan los canales de conexión. También se encarga de almacenar las estructuras necesarias para el envío posterior de información.

  Devuelve: -1 en caso de error, 0 si no.

- **int comedi_recibir(double * ret)**

  Realiza una lectura de la tarjeta.

  double * ret: almacena el valor leido

  Devuelve: -1 en caso de error, 0 si no.

- **int comedi_enviar(double valor)**

  Realiza un envío a la tarjeta.

  double valor: dato que será enviado a la tarjeta

  Devuelve: -1 en caso de error, 0 si no.

- **void comedi_cerrar(void)**

  Finaliza la conexión con la tarjeta DAQ

Realtime Manager

- **int realtime_on(void)**

  Convierte el programa que invoca a la función en una tarea en tiempo real. Puede modificarse la variable task_period_ns para definir el tiempo entre periodos que podrá utilizar.

  Devuelve: -1 en caso de error, 1 si no.

- **void realtime_wait(void)**

  Llamada a rt_task_wait_period() por si se desea establecer una nomenclatura común.

- **void realtime_off(void)**

  Llamada a rt_make_soft_real_time() por si se desea establecer una nomenclatura común.

Timeval Manager

- **void timeval_actual(struct timeval *tv)**

  Llamada a gettimeofday(tv, NULL) por si se desea establecer una nomenclatura común.

- **void timeval_set(struct timeval *tv, long sec, long usec)**

  Establece los valores de la estructura a lo indicado por los parámetros

  struct timeval *tv: estructura que almacena tiempos

  long sec: segundos a guardar

  long usec: microsegundo a guardar

- **void timeval_sleep(struct timeval *tv)**

  El programa se pausa el tiempo indicado por la estructura

  struct timeval *tv: tiempo a detener el programa

- **void min_sleep(void)**

  El programa duerme la mínima cantidad posible.

- **void nanosecs_sleep(int nanosecs)**

  El programa duerme los nanosegundos indicados.

  int nanosecs: nanosegundo a detener el programa.

- **void timeval_subtract(struct timeval *result, struct timeval *t1, struct timeval *t2)**

  Resta dos estructuras timeval si el resultado es mayor que cero. Si el resultado es menor que cero solo indica cero.

  struct timeval *result: estructura destino

  struct timeval *t1: minuendo

  struct timeval *t2: sustraendo

- **void timeval_sum(struct timeval *result, struct timeval *t1, struct timeval *t2)**

  Suma dos estructuras timeval.

  struct timeval *result: estructura destino

  struct timeval *t1: sumando a

  struct timeval *t2: sumando b

- **void timeval_div(struct timeval *t1, int n)**

  Divide una estructura timeval n veces.

  struct timeval *t1: estructura objetivo. Se sobreescribe con el resultado.

  int n: divisor

- **int timeval_mayor(struct timeval *t1, struct timeval *t2)**

  Comparador de estructuras timeval.

  struct timeval *t1: estructura a

  struct timeval *t2: estructura b

  Return: 1 si t1 es mayor o son iguales, 2 si es mayor t2

- **void timeval_copy(struct timeval *t1, struct timeval *t2)**

  Copia una estructura timeval.

  struct timeval *t1: estructura destino

  struct timeval *t2: estructura a copiar

- **void timeval_print(struct timeval *tv)**

  Imprime en segundos una estructura timeval.

  struct timeval *tv: estructura a imprimir

- **char * timeval_getStr(struct timeval *tv)**

  Proporciona una cadena de texto con el valor de una estructura timeval en segundos

  struct timeval *tv: estructura objetivo

  Return: cadena de texto con el valor de timeval en segundos

Models Tools

- **void ini_hr (double *x_ini, double *y_ini, double *z_ini, double *min, double *minB, double *max)**
- **void ini_iz (double *v_ini, double *u_ini, double *min, double *minABS, double *max)**
- **void ini_mr (double *x_ini, double *y_ini, double *min, double *minABS, double *max)**

  Estas funciones proporcionan puntos inicales para los modelos así como datos sobre sus puntos máximos y mínimos.

- **int ini_recibido (double *min, double *minABS, double *max)**

  Similar funcionalidad que las funciones ini_modelo(...) pero no proporciona datos de inicio y los máximos y mínimos son obtenidos por medio de la tarjeta DAQ. Utiliza la librería Comedi Manager para ello.

  Return: -1 en caso de error, 0 si no

- **void calcula_escala (double min_virtual, double max_virtual, double min_viva, double max_viva, double *escala_virtual_a_viva, double *escala_viva_a_virtual, double *offset_virtual_a_viva, double *offset_viva_a_virtual)**

  Calcula los valores necesarios para escalar dos neuronas/modelos

  double min_virtual: valor mínimo de un modelo a

  double max_virtual: valor máximo de un modelo a

  double min_viva: valor mínimo de un modelo b

  double max_viva: valor máximo de un modelo b

  double *escala_virtual_a_viva: variable donde se guardará la escala A->B

  double *escala_viva_a_virtual: variable donde se guardará la escala B->A

  double *offset_virtual_a_viva: variable donde se guardará el offset A->B

  double *offset_viva_a_virtual: variable donde se guardará el offset B->A